

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Политехнический институт
институт
Конструкторско-технологическое обеспечение машиностроительных
производств
кафедра

УТВЕРЖДАЮ

Заведующий кафедрой

М.П. Головин

инициалы, фамилия

подпись

«12» 06 2018 г.

БАКАЛАВРСКАЯ РАБОТА

09.03.01 Информатика и вычислительная техника
код – наименование направления

«Мобильное приложение для управления базой данных материалов»
тема

Руководитель

02.06.18
подпись, дата

ст. преподаватель КТОМП
должность, ученая степень

Д. И. Морозов
инициалы, фамилия

Выпускник

02.06.18
подпись, дата

Е. Е. Куксов
инициалы, фамилия

Красноярск 2018

ЗАДАНИЕ НА БАКАЛАВРСКУЮ РАБОТУ

Выдано студенту: Куксову Евгению Евгеньевичу

Группа: МТ14-09Б

Направление (специальность): 09.03.01 Информатика и вычислительная техника, профиль 09.03.01.06 Системы автоматизированного проектирования в машиностроении

Тема бакалаврской работы: Мобильное приложение для управления базой данных материалов

Утверждена приказом по университету № 3936 от 19.03.2018

Руководитель БР: Д. И. Морозов, ст. преподаватель

Перечень разделов БР:

1. Обзор
2. Архитектура
3. Реализация

Перечень созданного программного обеспечения:

1. Приложение для ОС Android
2. База данных SQLite

Руководитель БР  Д. И. Морозов
подпись, инициалы и фамилия

Задание принял к исполнению  Е. Е. Куксов
подпись, инициалы и фамилия студента

«20» Марта 2018 г.

РЕФЕРАТ

Выпускная квалификационная работа по теме «Мобильное приложение для управления базой данных материалов» содержит 52 страницы текстового документа, 20 рисунков, 10 таблиц, 14 использованных источников.

МОБИЛЬНОЕ ПРИЛОЖЕНИЕ ДЛЯ УПРАВЛЕНИЯ БАЗОЙ ДАННЫХ МАТЕРИАЛОВ

Объект – мобильное приложение.

Цель работы: создать приложение, которое позволяет просматривать и создавать список физических параметров материалов в виде, адаптированном для разных CAE-сред.

Задачи:

1. Создать базу данных материалов
2. Разработать алгоритмы управления базой данных
3. Программная реализация алгоритмов в виде приложения

В результате проделанной работы было разработано мобильное приложение под операционную систему Android, которое умеет создавать и просматривать список физических параметров материалов в виде, адаптированном для разных CAE-систем.

Оглавление	
ВВЕДЕНИЕ.....	5
1 Обзор.....	6
1.1 Технологии.....	6
1.1.1 Операционная система Android.....	6
1.1.2 Реляционные базы данных.....	7
1.1.3 Нормализация баз данных.....	8
1.1.4 SQLite.....	11
1.1.5 Android Studio.....	12
1.1.6 Жизненный цикл приложения Android.....	13
1.2 Обзор аналогов.....	13
2 Архитектура программного обеспечения.....	17
2.1 Требования к программному обеспечению.....	17
2.2 Модули приложения.....	18
2.3 Создание базы данных.....	19
2.4 Пользовательский интерфейс.....	24
3 Реализация.....	35
3.1 Заполнение базы данных.....	35
3.2 Модуль вывода параметров.....	37
3.3 Модуль создания нового материала.....	40
3.4 Модуль работы с созданными материалами.....	43
3.5 Модуль добавления параметров.....	45
3.6 Тестирование приложения.....	47
ВЫВОДЫ.....	50
СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ.....	51

ВВЕДЕНИЕ

Данная работа освещает создание и разработку мобильного приложения для управления базой данных материалов для CAE-сред [1], таких как Ansys, Solidworks Simulation разных версий и локализаций. Работая в этих программах пользователю иногда приходится создавать новые материалы, а их создание подразумевает заполнение списка параметров (например: модуль Юнга, плотность, коэффициент Пуассона, предел прочности, предел текучести). Цель приложения – облегчить этот процесс, а именно оптимизировать его и сэкономить пользователю большое количество времени. Это происходит за счёт того, что исключается процесс самостоятельного поиска всех данных для каждого нового материала, что особенно сложно для неопытных пользователей, например, студентов, проходящих обучение конечно-элементному анализу. Сложность заключается в том, что у различных версий CAE-программ перечень, порядок, терминология и единицы измерения параметров материалов могут существенно различаться.

1 Обзор

При разработке приложения задействовано много различных технологических ресурсов. Одними из таких являются:

- Настройки приложения
- Операционная система Android
- Реляционные базы данных
- Нормализация баз данных
- SQLite
- Android Studio

Далее рассмотрим каждый более подробно.

1.1 Технологии

1.1.1 Операционная система Android

Android — операционная система для смартфонов, планшетов, электронных книг, наручных часов, игровых приставок, нетбуков, очков Google [2], телевизоров и т.д [3]. В 2015 году появилась поддержка автомобильных развлекательных систем и бытовых роботов. Основана на ядре Linux и собственной реализации виртуальной машины Java от Google. Изначально разрабатывалась компанией Android, Inc., которую затем купила Google. Впоследствии Google инициировала создание альянса Open Handset Alliance (ОНА), который сейчас занимается поддержкой и дальнейшим развитием платформы. Android позволяет создавать Java-приложения, управляющие устройством через разработанные Google библиотеки.

У платформы Android есть свои API, (Application Programming Interface) - это интерфейс программирования, интерфейс создания приложений [4].

Создаваемое приложение будет работать на смартфонах и планшетах, где версия Android не ниже 4.1 (Jelly Bean) и версия API не ниже 16, так как, по

статистическим данным Google за 2017 год, версии Android-а ниже 4.1 уже практически неактуальны [5].

1.1.2 Реляционные базы данных

Реляционная база данных— это набор данных с предопределенными связями между ними [6]. Эти данные организованны в виде набора таблиц, состоящих из столбцов и строк. В таблицах хранится информация об объектах, представленных в базе данных. В каждом столбце таблицы хранится определенный тип данных, в каждой ячейке – значение атрибута. Каждая строка таблицы представляет собой набор связанных значений, относящихся к одному объекту или сущности. Каждая строка в таблице может быть помечена уникальным идентификатором, называемым первичным ключом, а строки из нескольких таблиц могут быть связаны с помощью внешних ключей. К этим данным можно получить доступ многими способами, и при этом реорганизовывать таблицы БД не требуется. Пример реляционной базы данных изображен на рисунке 1.1.

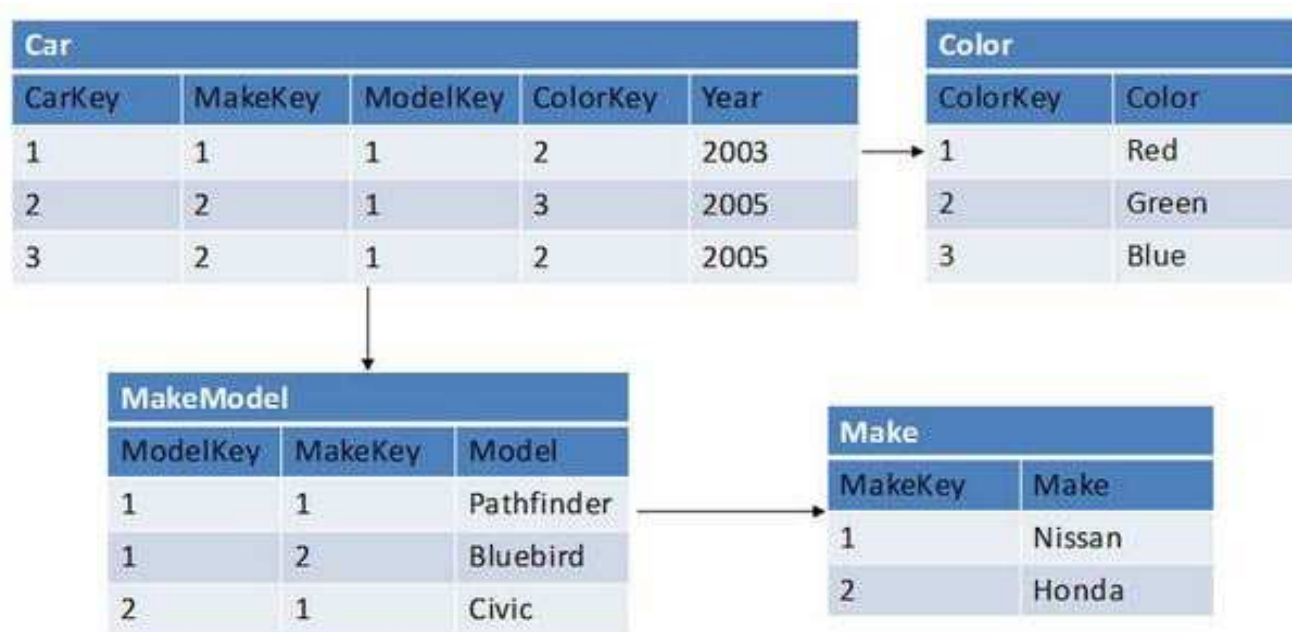


Рисунок 1.1 – Пример реляционной БД

1.1.3 Нормализация баз данных

Нормализация — это процесс организации данных в базе данных, включающий создание таблиц и установление отношений между ними в соответствии с правилами, которые обеспечивают защиту данных и делают базу данных более гибкой, устраняя избыточность и несогласованные зависимости [7].

Избыточность данных приводит к непродуктивному расходованию свободного места на диске и затрудняет обслуживание баз данных. Например, если данные, хранящиеся в нескольких местах, потребуется изменить, в них придется внести одни и те же изменения во всех этих местах. Изменение адреса клиента гораздо легче реализовать, если в базе данных эти сведения хранятся только в таблице Customers и нигде больше.

Что такое «несогласованные зависимости»? Пользователь, которому нужно узнать, например, адрес определенного клиента, вполне обоснованно будет искать его в таблице Customers (клиенты), но искать в ней сведения о зарплате сотрудника, который работает с этим клиентом, не имеет смысла. Зарплата сотрудника связана с сотрудником (зависит от него), поэтому эти сведения следует хранить в таблице Employees (сотрудники). Несогласованные зависимости могут затруднять доступ к данным, так как путь к данным при этом может отсутствовать или быть неправильным.

Существует несколько правил нормализации баз данных. Каждое правило называется «нормальной формой». Если выполняется первое правило, говорят, что база данных представлена в «первой нормальной форме». Если выполняются три первых правила, считается, что база данных представлена в «третьей нормальной форме». Есть и другие уровни нормализации, однако для большинства приложений достаточно нормализовать базы данных до третьей нормальной формы.

Как и в случае со многими другими формальными правилами и спецификациями, обеспечить полное соответствие реальным ситуациям не

всегда возможно. Как правило, для выполнения нормализации приходится создавать дополнительные таблицы, и некоторые клиенты считают это нежелательным. Собираясь нарушить одно из первых трех правил нормализации, убедитесь в том, что в приложении учтены все связанные с этим проблемы, такие как избыточность данных и несогласованные зависимости.

В описаниях ниже приведены соответствующие примеры.

Первая нормальная форма

- Устраните повторяющиеся группы в отдельных таблицах.
- Создайте отдельную таблицу для каждого набора связанных данных.
- Идентифицируйте каждый набор связанных данных с помощью первичного ключа.

Не используйте несколько полей в одной таблице для хранения похожих данных. Например, для слежения за товаром, который закупается у двух разных поставщиков, можно создать запись с полями, определяющими код первого поставщика и код второго поставщика.

Что произойдет при добавлении третьего поставщика? Добавление третьего поля нежелательно, так как для этого нужно изменять программу и таблицу, поэтому данный способ плохо адаптируется к динамическому изменению числа поставщиков. Вместо этого можно поместить все сведения о поставщиках в отдельную таблицу Vendors (поставщики) и связать товары с поставщиками с помощью кодов товаров или поставщиков с товарами с помощью кодов поставщиков.

Вторая нормальная форма

- Создайте отдельные таблицы для наборов значений, относящихся к нескольким записям.
- Свяжите эти таблицы с помощью внешнего ключа.

Записи могут зависеть только от первичного ключа таблицы (составного ключа, если необходимо). Возьмем для примера адрес клиента в системе бухгалтерского учета. Этот адрес необходим не только таблице Customers, но и таблицам Orders, Shipping, Invoices, Accounts Receivable и Collections. Вместо

того чтобы хранить адрес клиента как отдельный элемент в каждой из этих таблиц, храните его в одном месте: или в таблице Customers, или в отдельной таблице Addresses.

Третья нормальная форма

- Устраните поля, не зависящие от ключа.

Значения, входящие в запись и не являющиеся частью ключа этой записи, не принадлежат таблице. Если содержимое группы полей может относиться более чем к одной записи в таблице, подумайте о том, не поместить ли эти поля в отдельную таблицу.

Например, в таблицу Employee Recruitment (наем сотрудников) можно включить адрес кандидата и название университета, в котором он получил образование. Однако для организации групповой почтовой рассылки необходим полный список университетов. Если сведения об университетах будут храниться в таблице Candidates, составить список университетов при отсутствии кандидатов не получится. Таким образом, создайте вместо этого отдельную таблицу Universities и свяжите ее с таблицей Candidates при помощи ключа — кода университета.

Исключение. Выполнять нормализацию баз данных до третьей нормальной формы теоретически желательно, но не всегда практично. Например, для устранения всех возможных зависимостей между полями таблицы Customers придется создать отдельные таблицы для хранения сведений о городах, почтовых индексах, торговых представителях, категориях клиентов и любых других сведений, которые могут дублироваться в нескольких записях. С теоретической точки зрения нормализация желательна. Однако значительное увеличение числа маленьких таблиц может привести к снижению производительности СУБД или исчерпанию памяти и числа дескрипторов открытых файлов.

Выполнять нормализацию до третьей нормальной формы может быть целесообразно только для часто изменяемых данных. Если при этом сохранятся

зависимые поля, спроектируйте приложение так, чтобы при изменении одного из этих полей пользователь должен был проверить все связанные поля.

Другие формы нормализации

Кроме описанных нормальных форм, существует четвертая нормальная форма, которую также называют нормальной формой Бойса-Кодда (BCNF), и пятая нормальная форма, но на практике они применяются редко. Несоблюдение этих правил может привести к ухудшению архитектуры базы данных, но на функциональности это сказаться не должно.

1.1.4 SQLite

SQLite - это реляционная база данных, запросы к которой можно осуществлять при помощи языка запросов SQL [8].

Отличие SQLite от MySQL и аналогичных СУБД.

Классические СУБД, такие как MySQL (а также MS SQL, Oracle, PostgreSQL) состоят из отдельного сервера, поддерживающего работу базы данных и прослушивающих определённый порт, на предмет обращения клиентов. В качестве клиента может выступать в том числе и расширение PHP, реализующего интерфейс, с помощью которого осуществляются запросы к базе. Движок SQLite и интерфейс к ней реализованы в одной библиотеке, что увеличивает скорость выполнения запросов. Такой сервер часто называют встроенным.

Особенности SQLite.

Все базы данных хранятся в файлах, по одному файлу на базу. Количество баз данных, а также таблиц в них, ограничено только свободным местом, имеющимся на сайте. А максимально возможный объём одной базы данных составляет 2 Тб. Так как все данные хранятся в файлах, проблем с переносом базы данных с одного хостинга на другой не существует – достаточно лишь скопировать соответствующие файлы.

1.1.5 Android Studio

Android Studio — интегрированная среда разработки производства Google, с помощью которой разработчикам становятся доступны инструменты для создания приложений на платформе Android OS [9]. Android Studio можно установить на Windows, Mac и Linux. Учетная запись разработчика приложений в Google Play App Store стоит \$25. Android Studio создавалась на базе IntelliJ IDEA.

IDE можно загрузить и пользоваться бесплатно. В ней присутствуют макеты для создания UI, с чего обычно начинается работа над приложением. В Studio содержатся инструменты для разработки решений для смартфонов и планшетов, а также новые технологические решения для Android TV, Android Wear, Android Auto, Glass и дополнительные контекстуальные модули.

Среда Android Studio предназначена как для небольших команд разработчиков мобильных приложений (даже в количестве одного человека), или же крупных международных организаций с GIT или другими подобными системами управления версиями. Опытные разработчики смогут выбрать инструменты, которые больше подходят для масштабных проектов. Решения для Android разрабатываются в Android Studio с использованием Java или C++. В основе рабочего процесса Android Studio заложен концепт непрерывной интеграции, позволяющий сразу же обнаруживать имеющиеся проблемы. Продолжительная проверка кода обеспечивает возможность эффективной обратной связи с разработчиками. Такая опция позволяет быстрее опубликовать версию мобильного приложения в Google Play App Store. Для этого присутствует также поддержка инструментов LINT, Pro-Guard и App Signing.

С помощью средств оценки производительности определяется состояние файла с пакетом прикладных программ. Визуализация графики дает возможность узнать, соответствует ли приложение ориентир Google в 16 миллисекунд. С помощью инструмента для визуализации памяти разработчик узнает, когда его приложение будет использовать слишком много оперативной

памяти и когда произойдет «сборка мусора». Инструменты для анализа батареи показывают, какая нагрузка приходится на устройство.

Android Studio совместима с платформой Google App Engine для быстрой интеграции в облаке новых API и функций. В среде разработки вы найдете различные API, такие как Google Play, Android Pay и Health. Присутствует поддержка всех платформ Android, начиная с Android 1.6. Есть варианты Android, которые существенно отличаются от версии Google Android. Самая популярная из них — это Amazon Fire OS. В Android Studio можно создавать APK для этой ОС. Поддержка Android Studio ограничивается онлайн-форумами.

1.1.6 Жизненный цикл приложения Android

Для удобства пользователя в приложении будет реализована такая функция, как «Начать с места, где закончил в прошлый раз», формулировка очень тривиальная, но объясняет принцип самой функции как нельзя лучше. Для того, чтобы реализовать эту функцию, необходимо разобраться в Android-lifecycle (Жизненный цикл) – это различные состояния андроид приложений такие как:

- Активно,
- Приостановлено,
- Уничтожено и т.д.

Чтобы правильно применить их в приложении, обратимся к официальному источнику [13] андроид разработки от Google.com и разберемся в нем.

1.2 Обзор аналогов

Перед тем как разрабатывать приложение, необходимо осведомиться о наличии аналогов. Это даст представление о актуальности и, в целом, необходимости создания – так как есть вероятность того, что приложение уже

существует. Далее рассмотрим несколько приложений той тематики, что собираемся разработать.

Приложение-аналог №1

- Название – «Сплавы металлов. Марочник» [10]
- Требуемая версия Android – 4.1+
- Разработчик – EnTrainings (Россия, Санкт-Петербург)
- Кол-во установок – 1000 - 5000

Достоинства:

- Частично бесплатное
- Наличие поиска
- Наличие категорий материалов

Недостатки:

- Отсутствие представления параметров в формате САЕ-сред (например Ansys. Solidworks и т.д.)
- Частично бесплатное
- Нет возможности создавать свои материалы
- Нерасширяемая база данных

Это приложение предоставляет полный справочник металлов и сплавов: стали, меди, чугуна, свинца, драгоценных металлов и др.

В базе программы около 3000 марок.

Здесь вы найдете описание сплавов – их область применения, ГОСТы, химический состав.

Все металлы удобно структурированы по группам: алюминий, бронза, висмут, вольфрам, галлий, гафний йодидный, золото, индий, иридий, кадмий, кобальт, кремний, латунь, литий, магний, марганец, медь, молибден, никель, ниобий, олово, осмий, Палладий, платина, порошок, родий, ртуть, рутений, свинец, селен, серебро, сплавы (Fe, Ni, Cr,Co), сталь, сурьма, таллий, теллур, титан, ферросплав, хром, цинк, чугун — а также их подгруппы. Также вы можете найти любой сплав металла по марке.

Приложение-аналог №2

- Название – «Сплавы и Металлы» [11]
- Требуемая версия Android – 2.3+
- Разработчик - DronDon Developer (Украина, Донецкая область, Дзержинск)
- Кол-во установок – 50000 – 100000

Достоинства:

- Частично бесплатное приложение
- Интуитивно-понятный интерфейс
- Наличие поиска
- Наличие категорий материалов

Недостатки:

- Про-версия платная
- Слишком назойливая реклама
- Отсутствие представления параметров в формате САЕ-сред (например Ansys. Solidworks и т.д.)
- Нет возможности создавать свои материалы
- Нерасширяемая база данных

Программа предназначена для поиска металлов и сплавов по их названию или выбору по категориям.

В базе насчитывается больше 1500 металлов и сплавов, применяемых в металлургии.

Приложение-аналог №3

- Название – «Справочник инженера сварщика» [13]
- Требуемая версия Android – 4.0+
- Разработчик - Sergey Malyshev (Россия, Владивосток)
- Кол-во установок – 10000 – 50000

Достоинства:

- Бесплатное приложение
- Интуитивно-понятный интерфейс
- Наличие категорий материалов

Недостатки:

- Мало материалов
- Нет поиска по материалам
- Отсутствие представления параметров в формате САЕ-сред (например Ansys. Solidworks и т.д.)
- Нет возможности создавать свои материалы
- Нерасширяемая база данных

"Справочник инженера сварщика" разработан в помощь специалистам сварочного производства при предварительной оценке норм расхода сварочных материалов и выбора технологии сварки.

Содержит справочные материалы по технологии сварки стальных конструкций, расчету основных (по размерам и весу) и сварочных материалов (по весу проката и длинам швов), маркам сталей и сплавов (с указанием подходящих сварочных материалов).

Возможности приложения (Стандартные):

- Технология сварки стальных конструкций (разработана с учетом Рекомендаций САСв);
- Металлокалькулятор;
- Расчет сварочных материалов;
- Диаграмма Шеффлера;
- Марочник часто применяемых сталей и сплавов;
- Расчет параметров сварки.

2 Архитектура программного обеспечения

2.1 Требования к программному обеспечению

Требования к программному обеспечению были сформулированы исходя из поставленных задач и после обзора аналогов. После чего, на их основе было разработано программное обеспечение. Требования к программному обеспечению представлены в таблице 2.1.

Таблица 2.1 Требования к программному обеспечению.

Кому это нужно?	Что ему нужно?	Почему/Зачем ему это нужно?
Пользователь	Справочник гостовских материалов	Быстрый доступ к необходимым материалам, а в следствии - экономия времени при выполнении задания
Дипломный руководитель	Возможность добавления материалов в справочник	Чтобы была возможность расширения справочника
Программист	Поддержка Android начиная с версии 4.0 и API 16	Т.к. в версиях ниже не реализованы некоторые функции, которые необходимы при разработке
Дипломный руководитель	Закомментированный код	Это даст возможность легко разобраться в коде и в следствии чего можно будет дорабатывать и усовершенствовать приложение
Дипломный руководитель	Залить приложение на систему контроля версий Bitbucket	Для обеспечения безопасности (в случае непредвиденных обстоятельств со случайным удалением и т.д.), а так же удобство в наглядности изменений кода
Пользователь	Двухязычность	Общепринятые требования
Программист	Залить на Play Market	Для массового распространения приложения
Пользователь	Справочная система	Для решения возникающих вопросов при использовании приложения
Программист	Нормализация БД	Общепринятые требования коммерческих БД

Программист	Адаптивные иконки	Для обеспечения унификации на различных темах Android-устройств
Пользователь	Категории материалов	Для обеспечения удобной навигации и поиска
Программист, Пользователь	Жизненный цикл приложения	Чтобы приложение не вылетало с ошибкой
Программист, Пользователь	Поддержка разных разрешений	Чтобы использовать на разных устройствах
Программист	Поддержка только портретной ориентации	Для упрощения разработки

2.2 Модули приложения

Далее распишем назначение всех модулей, указанных на рисунке 2.1.

Модуль представления главного окна – этот модуль формирует интерфейс первичного Activity из xml-документа, которое пользователь видит при запуске приложения.

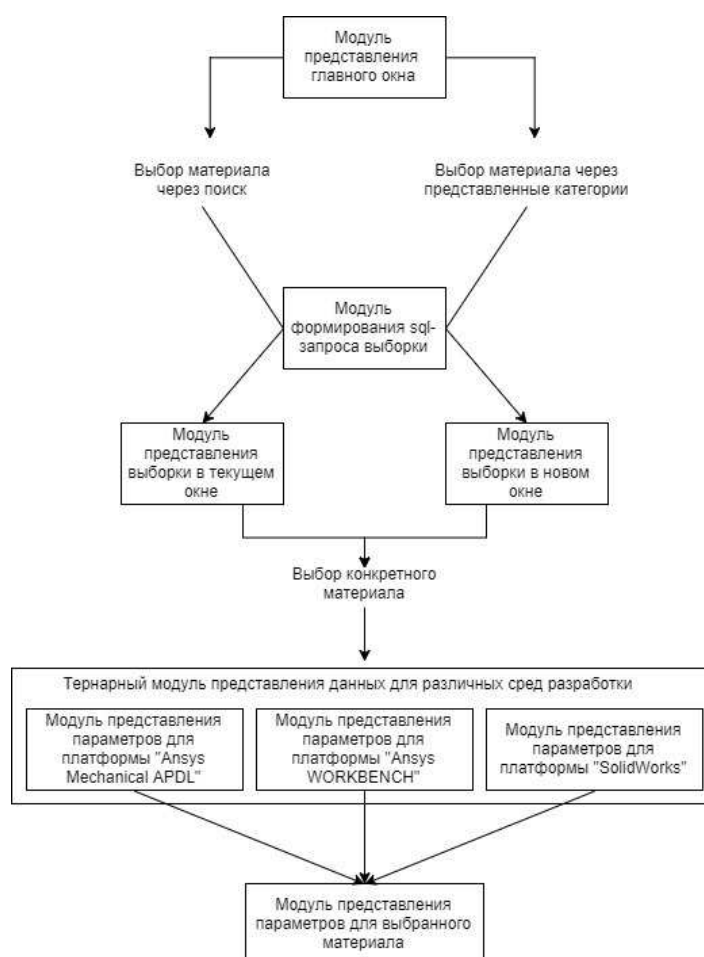


Рисунок 2.1 – Схема модулей приложения

На этом Activity предоставлены два варианта поиска материала:

1. Осуществление поиска через «строку поиска» (вверху Activity)
2. Осуществление поиска через представленные ниже «категории материалов»

Модуль формирования sql-запроса выборки – этот модуль формирует запрос в базу данных исходя из условий пользователя, а именно, если это поиск через «строку поиска» - то формирование запроса происходит на основе вписанных данных в эту строку. А если через «категории материалов» - то запрос формируется из названия нажатой категории.

Модуль представления выборки в текущем окне – этот модуль реализует представление результата выборки sql-запроса непосредственно под «строкой поиска», без создания и открытия нового Activity.

Модуль представления выборки в новом окне - этот модуль реализует представление результата выборки sql-запроса, после нажатия выбранной категории, в новом, созданном Activity.

Тернарный модуль представления данных для различных сред разработки – этот модуль начинает свою работу в тот момент, когда пользователь наживает (выбирает) тот материал, который ему нужен. Появляется всплывающее окно с тремя кнопками для CAE-Сред Ansys, Ansys Workbench и Solidworks соответственно.

Модуль представления данных для выбранного материала – В этом модуле реализуется создание из xml-файла интерфейса и его открытие с параметрами материала для конкретно-выбранной системы автоматизированного проектирования.

2.3 Создание базы данных

Для полноценной работы приложения была необходима база данных (далее БД) материалов, в котором у каждого материала было минимум 5 основных параметров: модуль Юнга, коэффициент Пуассона, предел прочности, предел текучести, плотность. Исходя из этой идеи была разработана

база данных из нескольких таблиц и различными связями между ними. Далее рассмотрим поподробней.

Первым делом была спроектирована схема базы данных. Это было сделано для того, чтоб иметь наглядное представление о составе БД и взаимосвязях между таблицами. На рисунке 2.2 представлена схема БД.

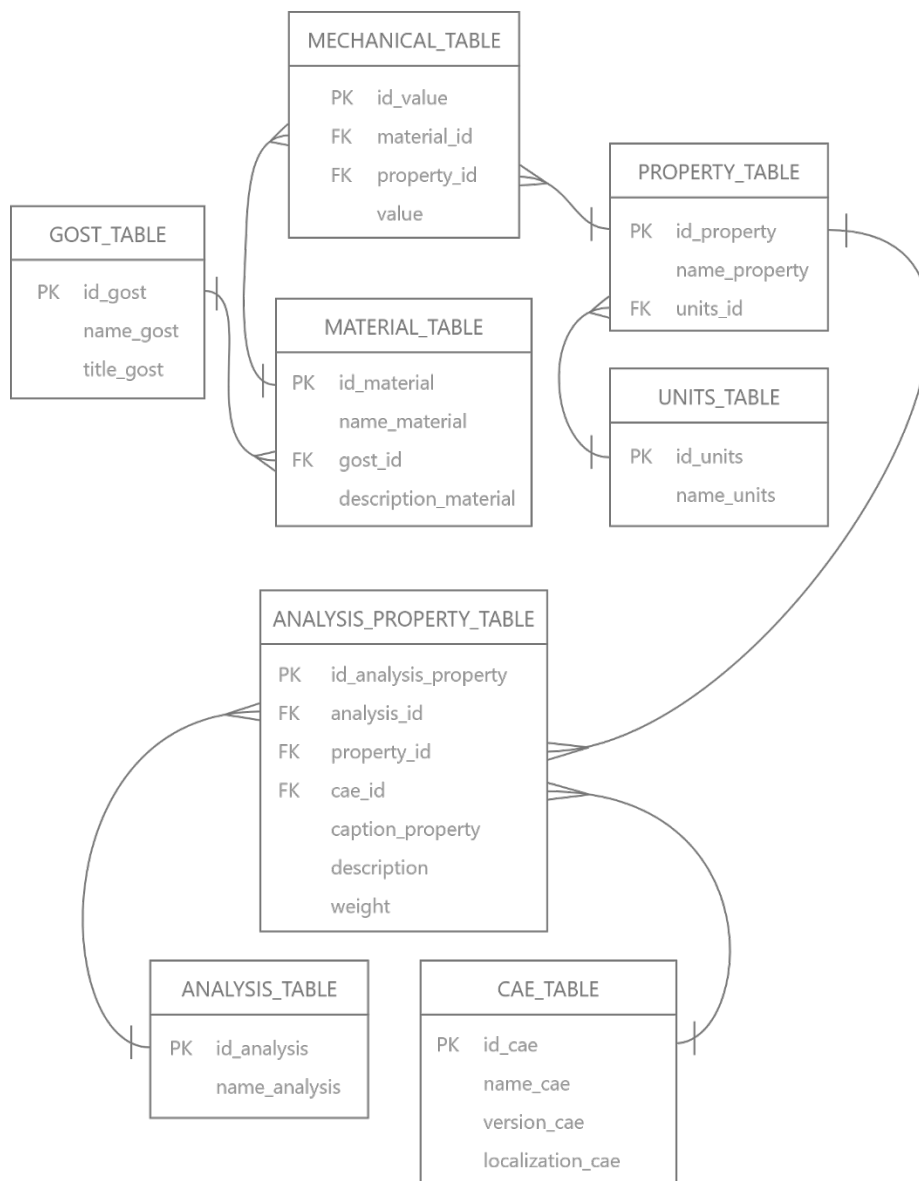


Рисунок 2.2 – Схема базы данных

Далее рассмотрим каждую таблицу детально.

Начнем с таблицы 2.2 - GOST_TABLE. В этой таблице хранятся соответствующие гости для каждого материала, находящегося в базе данных. В ней три столбца с именами id_gost, name_gost, title_gost. Столбец id_gost

является первичным ключом и обладает автоинкрементируемым заполнением с уникальностью данных.

Таблица 2.2 - GOST_TABLE

Имя столбца	Тип данных	Первичный ключ	Внешний ключ	Уникальность	Не NULL
id_gost	INTEGER	*		*	*
name_gost	TEXT				
title_gost	TEXT				

Следующая таблица 2.3 - MATERIAL_TABLE. В этой таблице хранятся названия всех материалов, находящихся в базе данных. В таблице присутствует связь один-ко-многим с таблицей GOST_TABLE, благодаря которой каждый материал имеет соответствующий ему ГОСТ. В Таблице четыре столбца с именами id_material, name_material, gost_id, description_material. Столбец id_material является первичным ключом и обладает автоинкрементируемым заполнением с уникальностью данных.

Таблица 2.3 - MATERIAL_TABLE

Имя столбца	Тип данных	Первичный ключ	Внешний ключ	Уникальность	Не NULL
id_material	INTEGER	*		*	*
name_material	TEXT				
gost_id	TEXT		id_gost		
descript_material	TEXT				

За ней идет таблица 2.4 - MECHANICAL_TABLE. Это одна из главных таблиц. В ней благодаря двум связям один-ко-многим с таблицами MATERIAL_TABLE и PROPERTY_TABLE, благодаря чему для каждого материала находятся 5 основных параметров с соответствующими значениями. В Таблице четыре столбца с именами id_value, material_id, property_id, value. Столбец id_value является первичным ключом и обладает автоинкрементируемым заполнением с уникальностью данных.

Таблица 2.4 – MECHANICAL_TABLE

Имя столбца	Тип данных	Первичный ключ	Внешний ключ	Уникальность	Не NULL

id_value	INTEGER	*		*	*
material_id	INTEGER		id_material		
property_id	INTEGER		id_property		
value	REAL				

Последующая таблица 2.5 - PROPERTY_TABLE. Эта таблица отвечает за хранение свойств для каждого материала. В ней присутствует связь один-ко-многим с таблицей UNITS_TABLE, благодаря чему для каждого свойства есть соответствующие ему единицы измерения. Столбец id_property является первичным ключом и обладает автоинкрементируемым заполнением с уникальностью данных.

Таблица 2.5 - PROPERTY_TABLE

Имя столбца	Тип данных	Первичный ключ	Внешний ключ	Уникальность	Не NULL
id_property	INTEGER	*		*	*
name_property	TEXT				
units_id	INTEGER		id_units		

После нее идет таблица 2.6 - UNITS_TABLE. Эта таблица отвечает за хранение единиц измерения. Столбец id_units является первичным ключом и обладает автоинкрементируемым заполнением с уникальностью данных.

Таблица 2.6 - UNITS_TABLE

Имя столбца	Тип данных	Первичный ключ	Внешний ключ	Уникальность	Не NULL
id_units	INTEGER	*		*	*
name_units	TEXT				

Затем таблица 2.7 - ANALYSIS_PROPERTY_TABLE. Эта еще одна главная таблица. В ней присутствуют три связи один-ко-многим с таблицами PROPERTY_TABLE, CAE_TABLE и ANALYSIS_TABLE. Основная цель этой таблицы разграничить описания 5 основных параметров в зависимости от CAE-среды и располагать их в нужной последовательности, в зависимости от того, какая она в конкретной CAE-среде. Столбец id_analysis_property является

первичным ключом и обладает автоинкрементируемым заполнением с уникальностью данных.

Таблица 2.7 - ANALYSIS_PROPERTY_TABLE

Имя столбца	Тип данных	Первичный ключ	Внешний ключ	Уникальность	He NULL
id_analysis_property	INTEGER	*		*	*
analysis_id	INTEGER		id_analysis		
property_id	INTEGER		id_property		
cae_id	INTEGER		id_cae		
caption_property	TEXT				
description	TEXT				
weight	REAL				

Таблица 2.8 - CAE_TABLE. Эта таблица отвечает за хранение информации о CAE-средах, например, их версии и локализации. Имеет связь с таблицей ANALYSIS_PROPERTY_TABLE вида один-ко-многим. Столбец id_cae является первичным ключом и обладает автоинкрементируемым заполнением с уникальностью данных.

Таблица 2.8 - CAE_TABLE

Имя столбца	Тип данных	Первичный ключ	Внешний ключ	Уникальность	He NULL
id_cae	INTEGER	*		*	*
name_cae	TEXT				
version_cae	INTEGER				
localization_cae	TEXT				

И наконец таблица 2.9 - ANALYSIS_TABLE. Эта таблица отвечает за хранение информации о видах анализов проводимых CAE-средами, например, структурный или термальный. На данном этапе разработки не была использована, ее пригодность рассматривается в перспективе. Имеет связь с таблицей ANALYSIS_PROPERTY_TABLE вида один-ко-многим. Столбец

id_cae является первичным ключом и обладает автоинкрементируемым заполнением с уникальностью данных.

Таблица 2.9 - ANALYSIS_TABLE

Имя столбца	Тип данных	Первичный ключ	Внешний ключ	Уникальность	Не NULL
id_analysis	INTEGER	*		*	*
name_analysiss	TEXT				

2.4 Пользовательский интерфейс

В этом пункте описывается весь интерфейс приложения и методы взаимодействия пользователя с ним. Задача интерфейса - многократно упростить поиск данных в БД и сделать приложение действительно нужным и эффективным.

Приложение, на данном этапе, разрабатывается для платформы Android, а значит будет использоваться на смартфонах, чьи экраны не позволяют разместить весь функционал в одном окне. В следствии этого мы должны сделать его многооконным. Далее рассмотрим эти окна поподробнее.

Начнем с главного экрана, который видит пользователь при входе в приложение, рисунок 2.3. Позицией 1 обозначено «Боковое меню» приложения. Позицией 2 обозначены кнопки-категории, их назначение – это быстрый доступ к конкретной категории материалов, который непосредственно на них же и написан. Позиция 3 – это «Быстрый поиск», введя туда название какого-либо материала (если таковой имеется в базе) вы получите результат под этой строкой поиска, то есть кнопки-категории будут скрыты и на их месте появятся результаты поиска. Позиция 4 – это кнопка «Создание нового материала», нажав на нее откроется окно, в котором создается материал.

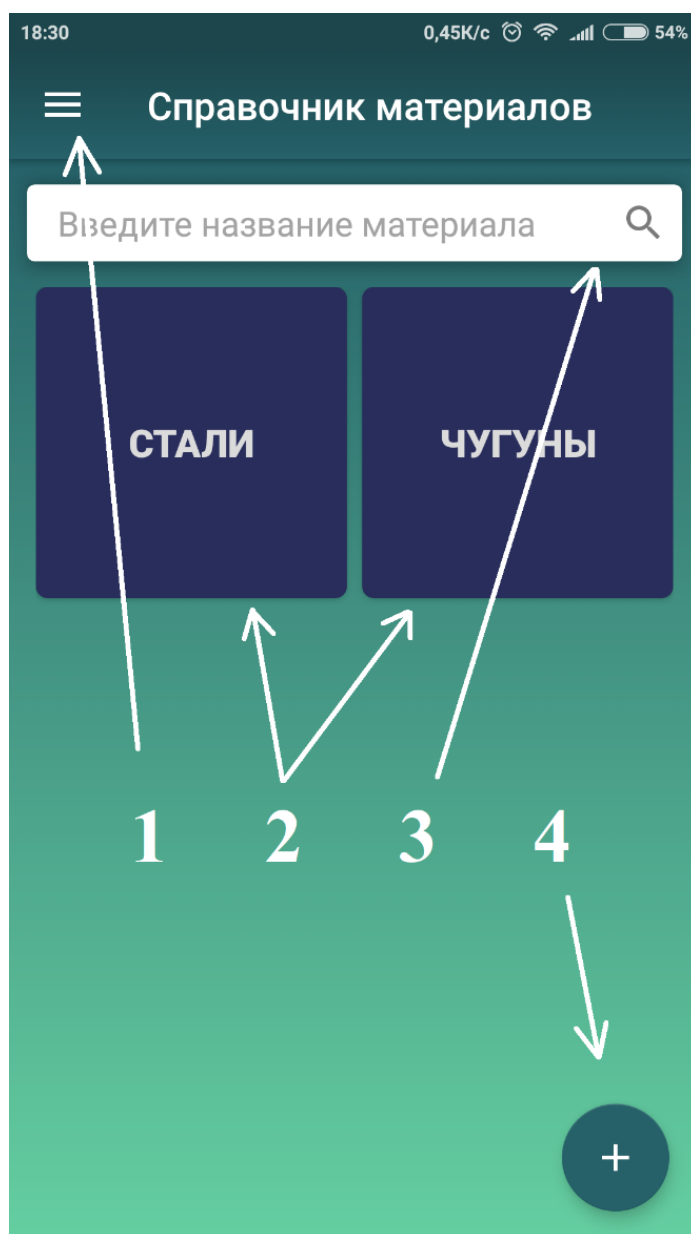


Рисунок 2.3 – Интерфейс главного окна

Следующее окно – это «Боковое меню», рисунок 2.4, здесь там находится раздел «Мои материалы» и описание версии приложения, в перспективе тут можно располагать все дополнительные, создаваемые разделы, для быстрого доступа к ним

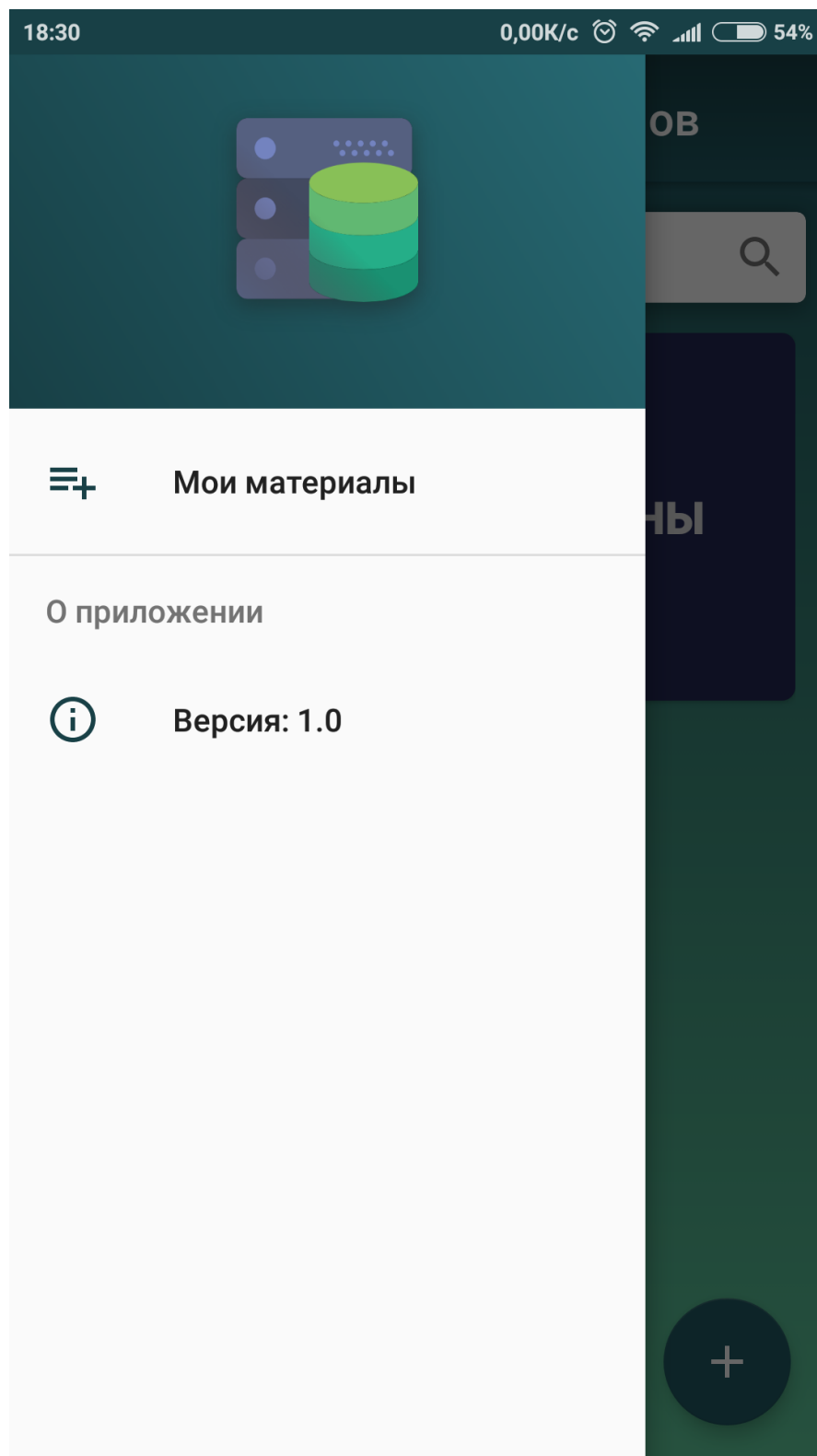


Рисунок 2.4 – Интерфейс бокового меню

Если нажать в «Боковом меню» на пункт «Мои материалы», то нас перебросит в хранилище материалов, созданных пользователем, рисунок 2.5. Здесь осуществляется просмотр, редактирование и удаление только тех материалов, которые были созданы сами пользователем, и только их.

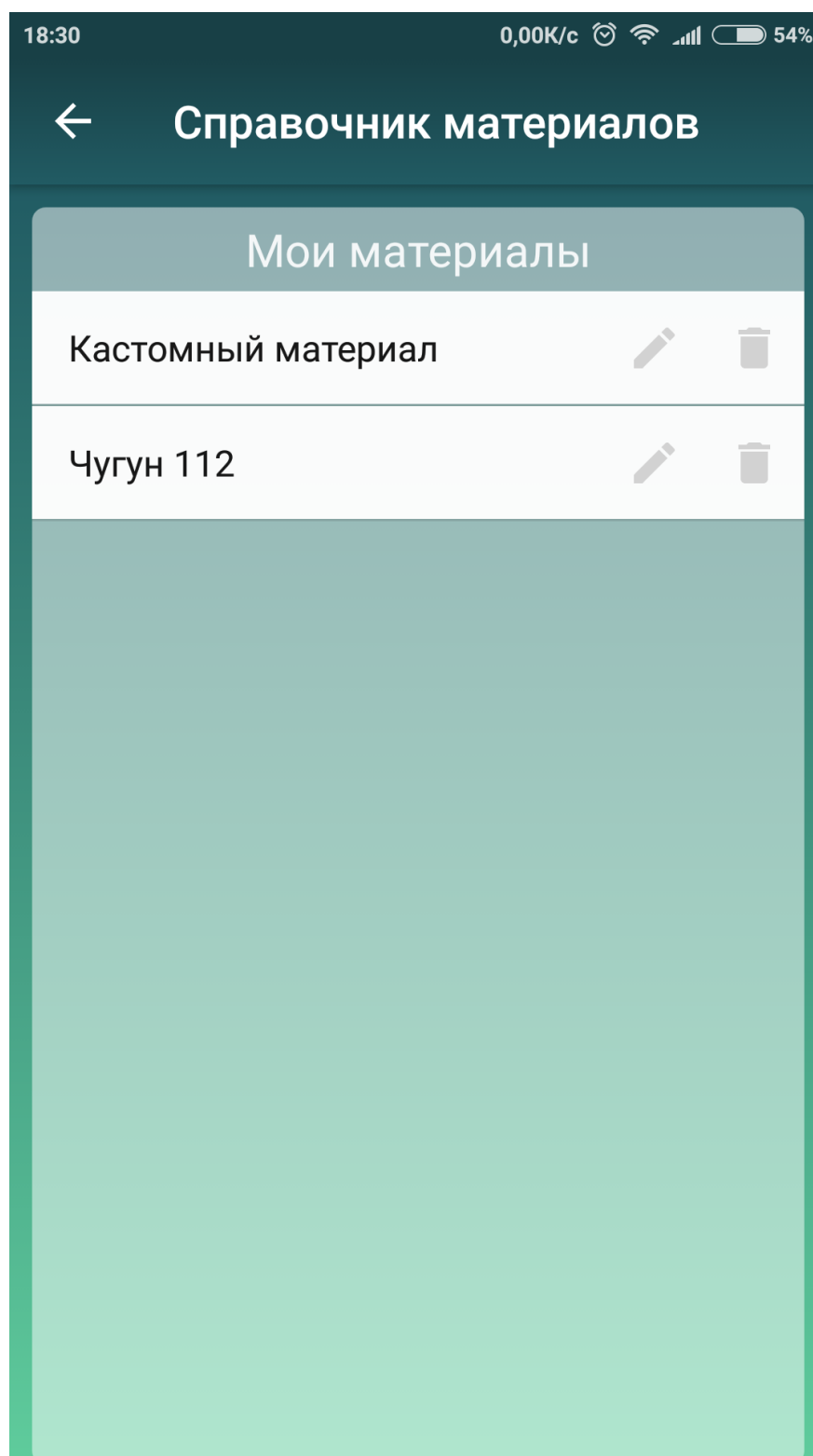


Рисунок 2.5 – Интерфейс раздела «Мои материалы»

Следующее окно – это результат нажатия на кнопку-категорию «Стали», которая располагается на главном экране. Здесь мы видим лишь список всех имеющихся сталей и их общее количество в верху экрана, рисунок 2.6. Аналогично происходит и с остальными кнопками-категориям: Нажимаем, к

примеру, «Чугуны» - приложение выводит нам все материалы с название чугуна, что есть в базе данных.

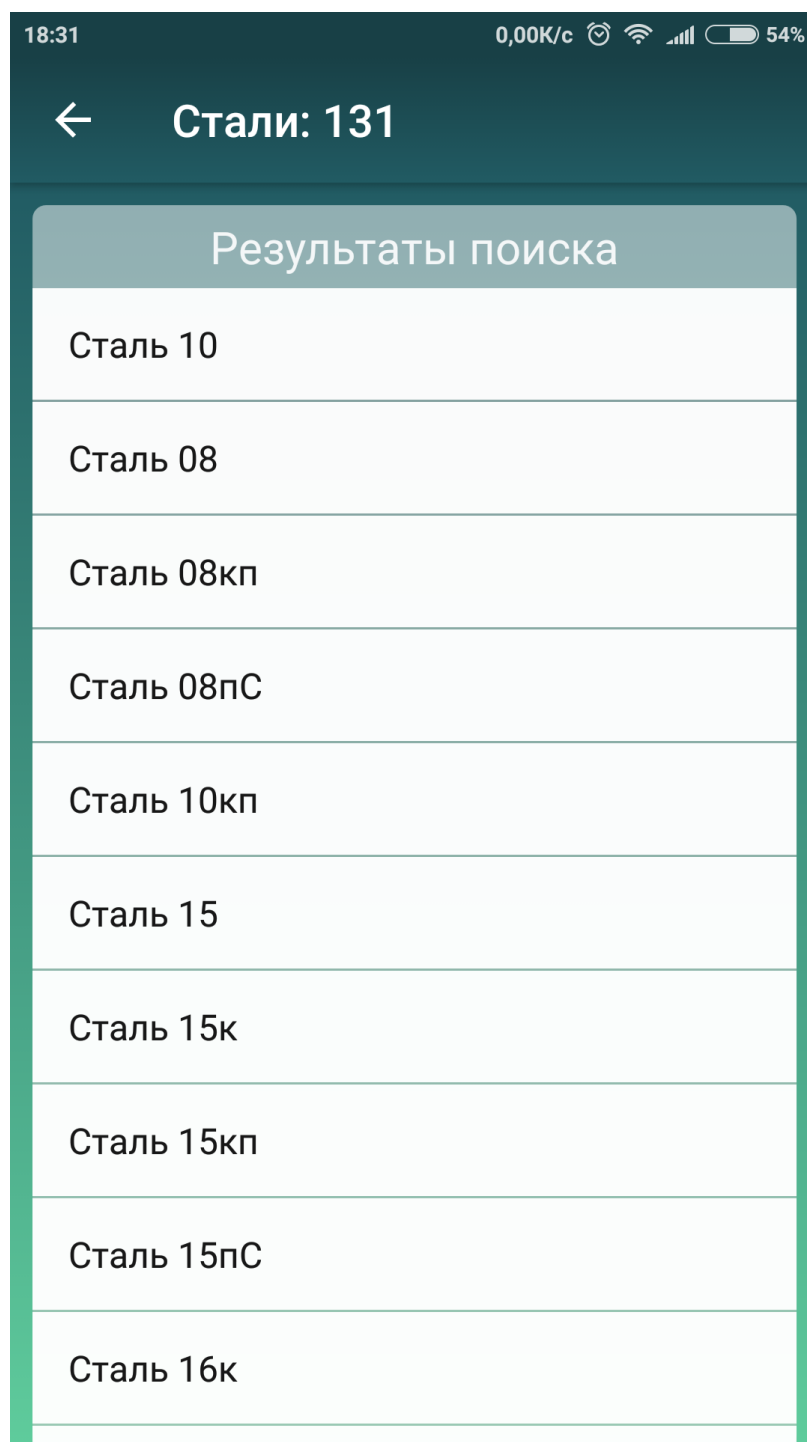


Рисунок 2.6 – Интерфейс окна по нажатию кнопки-категории

На рисунке 2.7 нам представлен интерфейс взаимодействия с динамическим поиском на главном окне. Принцип действия очень простой: ставим курсор в окно поиска, набираем какое-нибудь название материала, к примеру “сталь 15” и приложение выводит нам все материалы, в названии

которых есть совпадение с нашим запросом. Через эту строку поиска происходит поиск по всем, существующим в базе данных, материалам.

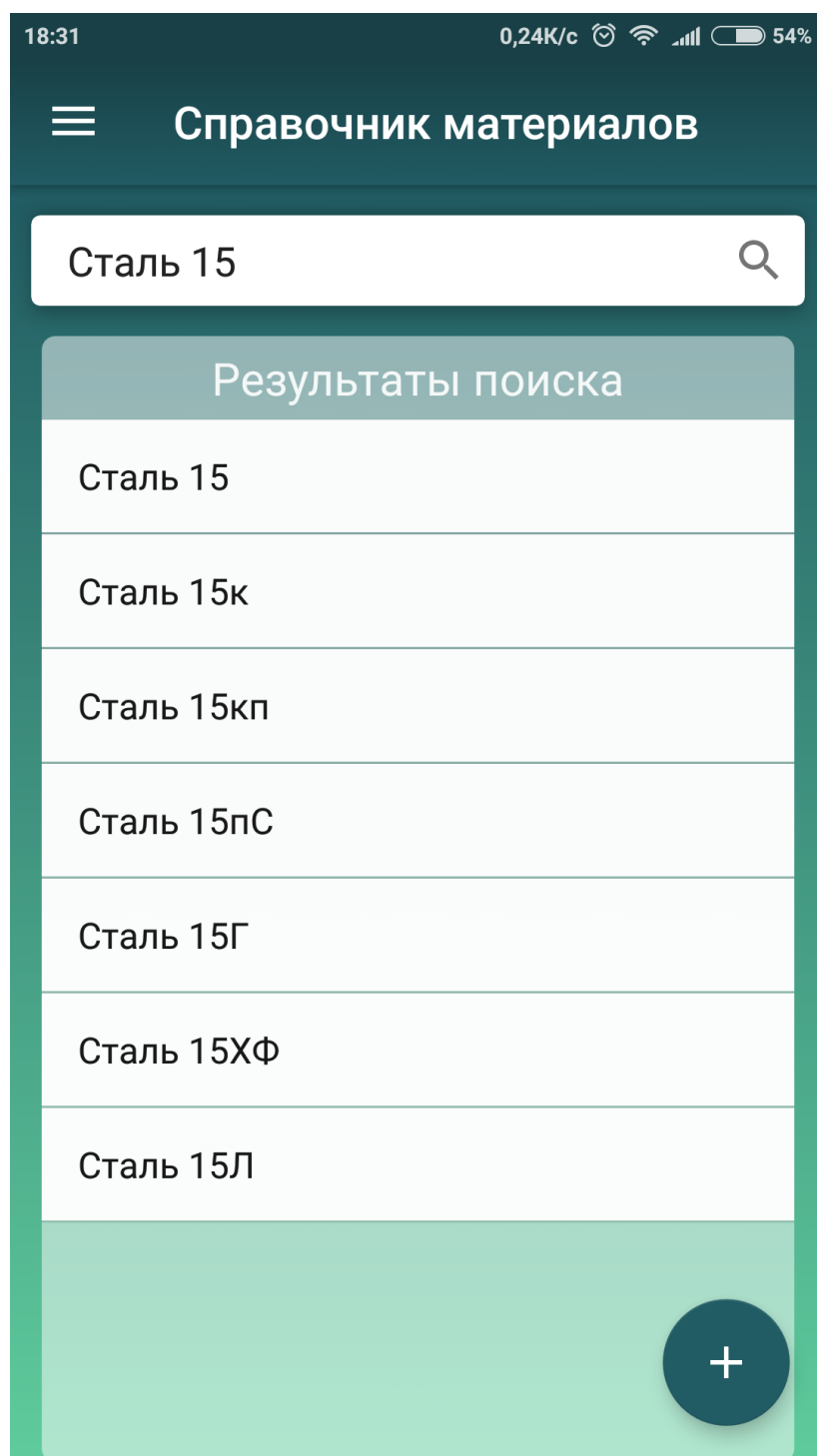


Рисунок 2.7 – Интерфейс динамического поиска

Далее рассмотрим то, что происходит в момент нажатия на какой-либо материал. Появляется всплывающее окно с выбором из трех САЕ-сред, а именно Ansys, Ansys Workbench и Solidworks, рисунок 2.8. При нажатии на кнопку с нужной нам САЕ-средой, нас перебросит в окно, где представлены 5

основных параметров (о которых говорилось выше, Модуль Юнга, Плотность и т.д.) в формате именно выбранной САЕ-среды.

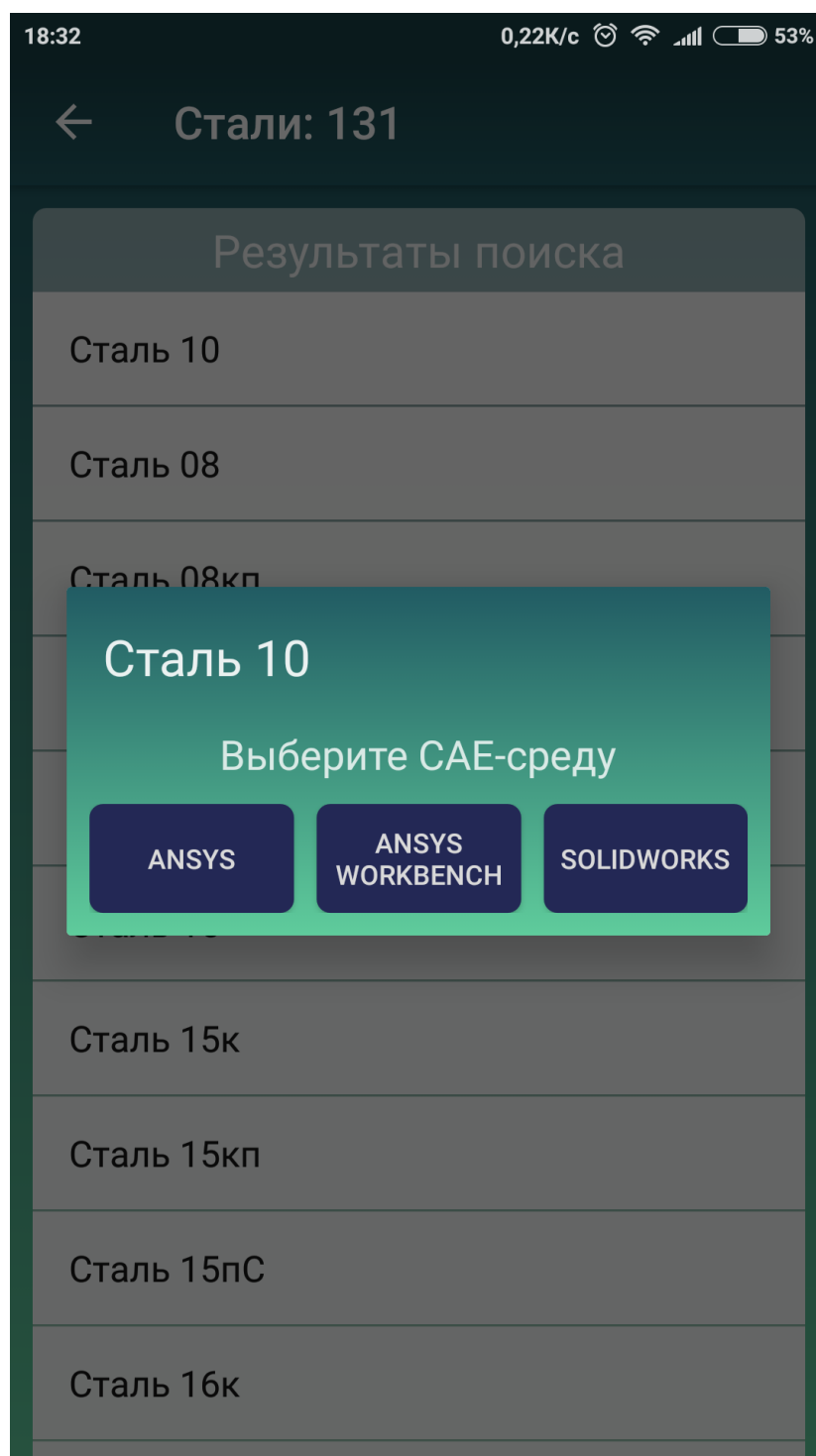


Рисунок 2.8 – Интерфейс всплывающего окна с выбором САЕ-среды

После того, как мы нажмем, на одну из кнопок с САЕ-средой, к примеру, на «Ansys», рисунок 2.9, то приложение откроет нам окно параметров материала, который мы выбрали. Но главная особенность в том, что эти параметры и называются, и расположены в таком же порядке, и имеют единицы

измерения, такие же, как и в самой программе Ansys. Сделано это, чтоб многократно упростить пользователю зрительное взаимодействие выбранного материала и САЕ-среды, в которой он работает.

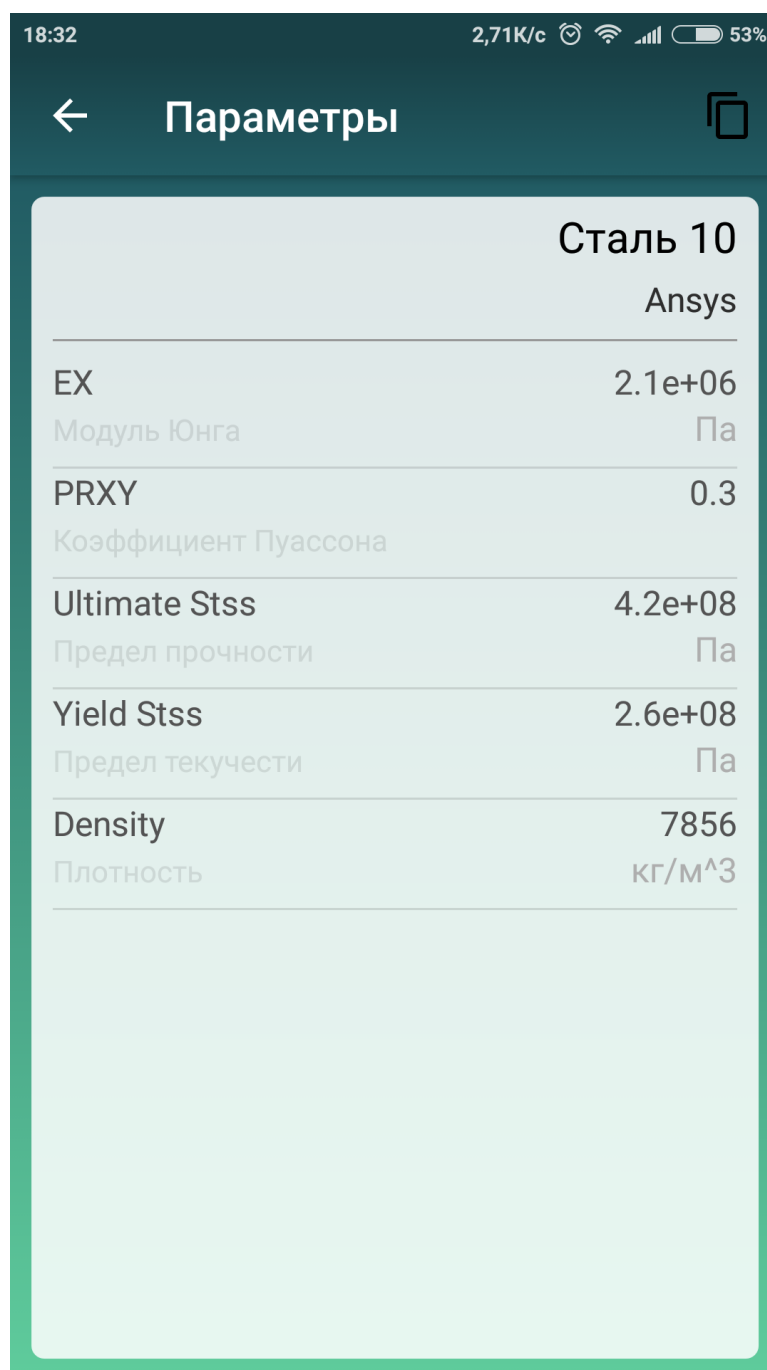


Рисунок 2.9 – Интерфейс параметров материала

На следующем рисунке 2.10 изображен интерфейс создания нового материала. Попадаем мы сюда по нажатию главного экрана под позицией 4. Здесь мы видим две карточки. В самой верхней вводится название создаваемого материала. Далее идет настраиваемый список параметров. Первые пять

параметров основные и должны быть в любом материале. Помимо этого, можно добавить свой параметр, нажав на кнопку со знаком плюс, внизу списка. После нажатия появится окошко для ввода названия, единиц измерения и значения нового параметра. Далее нажимаем ОК и параметр добавляется в список.

После того, как заполним все данные и список параметров полностью настроен, нажимаем на круглую кнопку с галочкой, это сохранение. Созданный материал сохранится в раздел «Мои материалы» в боковом меню, окно создания закроется и нас перекинет на главный экран.

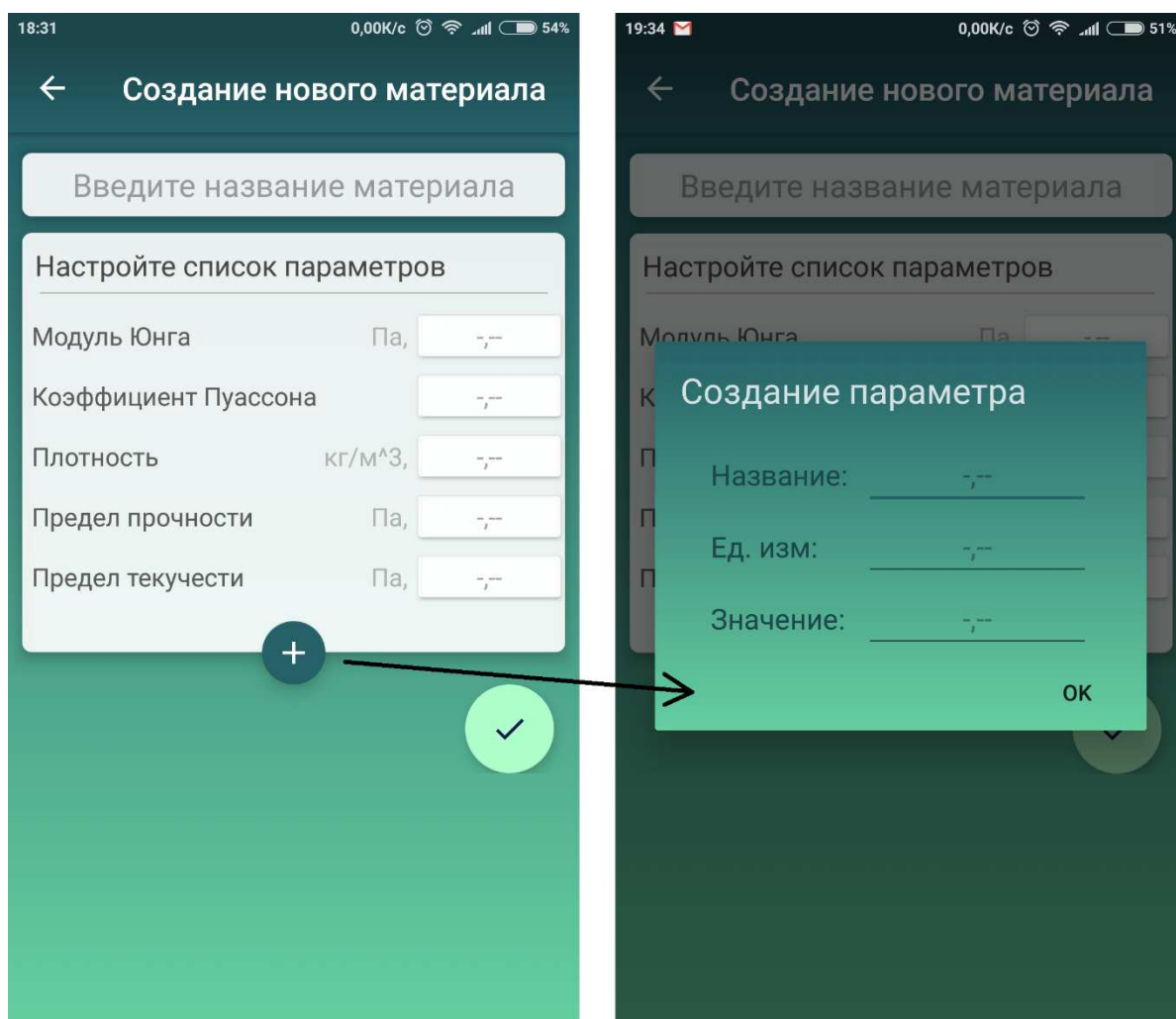


Рисунок 2.10 – Интерфейс создания материала и настройки списка параметров

Далее, на рисунке 2.11, мы видим окно создания копии материала. Делается это следующим образом: например, нам понравился какой-нибудь материал в базе, но хотелось бы расширить его параметры, к примеру, добавить температуру или массу. Мы открываем нужный нам материал и нажимаем

кнопку «Копировать», которая находится в правом верхнем углу приложения, нас перебрасывает на экран, где мы видим все параметры копируемого материала. Нам лишь осталось добавить свои и сохранить его.

18:32 0,00K/c 53%

← Создание копии материала

Сталь 10

Настройте список параметров

Модуль Юнга	Па,	2.1e+06
Коэффициент Пуассона		0.3
Плотность	кг/м ³ ,	7856
Предел прочности	Па,	4.2e+08
Предел текучести	Па,	2.6e+08

+

✓

Рисунок 2.11 – Интерфейс создания копии материала и настройки списка

Ну и на последнем рисунке мы видим работу функции «перевод параметров для САЕ-среды Solidworks». Реализована исходя из того, что в интернете мало видео уроков по работе в Solidworks-е на русском языке, они в основном английские и соответственно названия параметров в них тоже на

английском. Здесь и возникла надобность иметь возможно быстрого перевода указанных параметров на английский язык, что и показано на рисунке 2.12.

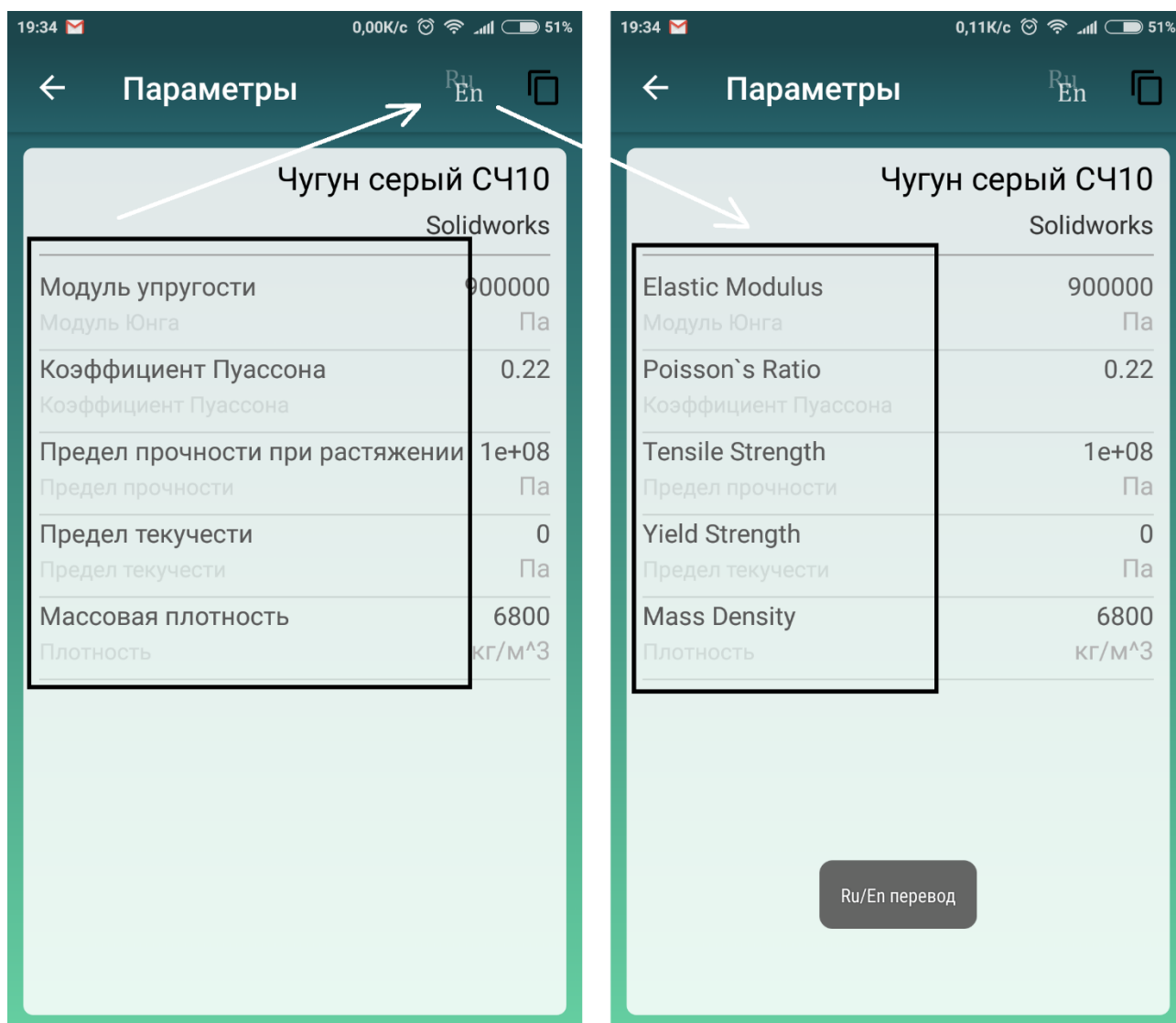


Рисунок 2.12 – Интерфейс быстрого перевода параметров на английский язык для CAE-среды Solidworks

3 Реализация

3.1 Заполнение базы данных

После создания базы данных необходимо наполнить ее материалами. Так как изначально не было какой-то примерной базы с материалами и их параметрами, хотя бы частично, то пришлось искать значения таких параметров как модуль Юнга, коэффициент Пуассона, предел прочности, предел текучести и плотность самостоятельно в сети интернет, а затем вручную переносить их в подготовленную базу. Делалось это следующим образом.

Так как вставлять значения пяти параметров для каждого материала слишком долго и трудоемко, а помимо того, можно еще и ошибиться, и ввести не те данные, то было придумано следующее решение. Создается Excel-документ и нем построчно для каждого материала записываются параметры, как показано на рисунке 3.1.

Марка	Гост (ru)	Е, Па	ν	ρ , кг/м ³	pred_tekuch, Па	pred_proch, Па
Сталь 10	ГОСТ 1050-88	2100000	0.3	7856	260000000	420000000
Сталь 08	ГОСТ 1050-88	2030000	0.3	7871	315000000	175000000
Сталь 08кп	ГОСТ 2590-2006	2030000	0.3	7871	175000000	295000000
Сталь 08пс	ГОСТ 2590-2006	2030000	0.3	7871	175000000	295000000
Сталь 10кп	ГОСТ 10702-78	1860000	0.3	7856	260000000	420000000
Сталь 15	ГОСТ 1050-88	2010000	0.3	7850	215000000	420000000
Сталь 15к	ГОСТ 5520-79	2110000	0.3	7870	225000000	370000000
Сталь 15кп	ГОСТ 2590-2006	2010000	0.3	7850	0	0
Сталь 15пс	ГОСТ 2590-2006	2010000	0.3	7850	0	0
Сталь 16к	ГОСТ 5520-79	0	0.3	0	255000000	400000000
Сталь 18к	ГОСТ 5520-79	0	0.3	0	275000000	430000000
Сталь 18кп	ГОСТ 23570-79	0	0.3	0	225000000	365000000
Сталь 20	ГОСТ 1050-88	2120000	0.3	7859	280000000	430000000
Сталь 20к	ГОСТ 5520-79	2110000	0.3	7870	245000000	400000000
Сталь 20кп	ГОСТ 2590-2006	2120000	0.3	0	295000000	490000000
Сталь 20пс	ГОСТ 2590-2006	2120000	0.3	0	295000000	490000000
Сталь 22к	ГОСТ 5520-79	2110000	0.3	7859	260000000	430000000
Сталь 25	ГОСТ 1050-88	1980000	0.3	7820	345000000	550000000
Сталь 30	ГОСТ 1050-88	2000000	0.3	7850	290000000	490000000
Сталь 35	ГОСТ 1050-88	2030000	0.3	7826	300000000	580000000
Сталь 40	ГОСТ 1050-88	2130000	0.3	7850	990000000	140000000
Сталь 45	ГОСТ 1050-88	2000000	0.3	7826	245000000	470000000
Сталь 50	ГОСТ 1050-88	2160000	0.3	7810	600000000	830000000
Сталь 55	ГОСТ 1050-88	2100000	0.3	7280	870000000	1060000000
Сталь 58	ГОСТ 1050-88	0	0.3	0	315000000	600000000

Рисунок 3.1 – Составление таблицы значений в Excel

После этого, так как ГОСТ-ы в основном повторяются, вручную переносим их уже в базу данных и затем проставляем id_значение строки каждого ГОСТ-а каждому материалу, в котором этот ГОСТ есть, как показано на рисунке 3.2.

id_gost	Марка	Гост (ru)	Е, Па	v	p, кг/м³	pred_tekuch, Па	pred_proch, Па
2	Сталь 10	ГОСТ 1050-88	2100000	0.3	7856	260000000	420000000
2	Сталь 08	ГОСТ 1050-88	2030000	0.3	7871	315000000	175000000
36	Сталь 08кп	ГОСТ 2590-2006	2030000	0.3	7871	175000000	295000000
36	Сталь 08пс	ГОСТ 2590-2006	2030000	0.3	7871	175000000	295000000
37	Сталь 10кп	ГОСТ 10702-78	1860000	0.3	7856	260000000	420000000
2	Сталь 15	ГОСТ 1050-88	2010000	0.3	7850	215000000	420000000
38	Сталь 15к	ГОСТ 5520-79	2110000	0.3	7870	225000000	370000000
36	Сталь 15кп	ГОСТ 2590-2006	2010000	0.3	7850	0	0
36	Сталь 15пс	ГОСТ 2590-2006	2010000	0.3	7850	0	0
38	Сталь 16к	ГОСТ 5520-79	0	0.3	0	255000000	400000000
38	Сталь 18к	ГОСТ 5520-79	0	0.3	0	275000000	430000000
39	Сталь 18кп	ГОСТ 23570-79	0	0.3	0	225000000	365000000
2	Сталь 20	ГОСТ 1050-88	2120000	0.3	7859	280000000	430000000
38	Сталь 20к	ГОСТ 5520-79	2110000	0.3	7870	245000000	400000000
36	Сталь 20кп	ГОСТ 2590-2006	2120000	0.3	0	295000000	490000000
36	Сталь 20пс	ГОСТ 2590-2006	2120000	0.3	0	295000000	490000000
38	Сталь 22к	ГОСТ 5520-79	2110000	0.3	7859	260000000	430000000
2	Сталь 25	ГОСТ 1050-88	1980000	0.3	7820	345000000	550000000
2	Сталь 30	ГОСТ 1050-88	2000000	0.3	7850	290000000	490000000
2	Сталь 35	ГОСТ 1050-88	2030000	0.3	7826	300000000	580000000
2	Сталь 40	ГОСТ 1050-88	2130000	0.3	7850	990000000	140000000
2	Сталь 45	ГОСТ 1050-88	2000000	0.3	7826	245000000	470000000
2	Сталь 50	ГОСТ 1050-88	2160000	0.3	7810	600000000	830000000
2	Сталь 55	ГОСТ 1050-88	2100000	0.3	7280	870000000	1060000000
2	Сталь 58	ГОСТ 1050-88	0	0.3	0	315000000	600000000

Рисунок 3.2 – Добавление столбца id из базы данных

Следующим шагом, заносим уже все данные, скопом, непосредственно в базу. Делается это при помощи вот таких запросов, показанных на рисунке 3.3

Добавление материалов в таблицу MATERIAL_TABLE			
INSERT INTO MATERIAL_TABLE (name_material, gost_id) VALUES("Сталь 10"	,	2);
INSERT INTO MATERIAL_TABLE (name_material, gost_id) VALUES("Сталь 08"	,	2);
INSERT INTO MATERIAL_TABLE (name_material, gost_id) VALUES("Сталь 08кп"	,	36);
INSERT INTO MATERIAL_TABLE (name_material, gost_id) VALUES("Сталь 08пс"	,	36);

Добавление параметров материалов в таблицу MECHANICAL_TABLE			
INSERT INTO MECHANICAL_TABLE (material_id, property_id, value) VALUES(33	,	2, 0.3);
INSERT INTO MECHANICAL_TABLE (material_id, property_id, value) VALUES(34	,	2, 0.3);
INSERT INTO MECHANICAL_TABLE (material_id, property_id, value) VALUES(35	,	2, 0.3);
INSERT INTO MECHANICAL_TABLE (material_id, property_id, value) VALUES(36	,	2, 0.3);

Рисунок 3.3 – SQL-запросы для добавления данных в базу

Эти запросы, затем, вставляются в программу SQLite Studio (в ней была создана и редактируется база данных), как показано на рисунке 3.4, она их обрабатывает и вносит данные в базу, распределяя их по указанным в запросе таблицам.

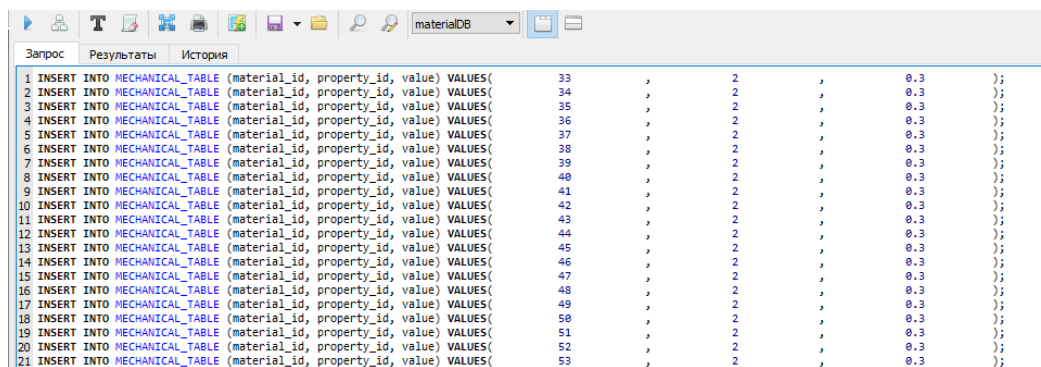


Рисунок 3.4 – Внесение SQL-запросов в программу

3.2 Модуль вывода параметров

Этот модуль приложения отвечает за представление параметров любого из материалов в таком в таком виде, каком они представлены в САЕ-средах. На вход принимает id материала и САЕ-среды, подставляет их в sql-запрос в блок условия WHERE и на их основе осуществляет выборку параметров, их описаний и единицы измерения из базы данных. Данный модуль реализован в виде Java-класса и содержит следующие методы:

```
//Создание экрана из xml-разметки
protected void onCreate(Bundle savedInstanceState)
{
    //Обработчик пунктов меню
    public boolean
    onOptionsItemSelected(android.view.MenuItem item)
    {
        //Создание меню
        public boolean onCreateOptionsMenu(Menu menu)
        {
            //Инициализация toolbar (панель инструментов)
            public void initToolbar()
            {
                //Инициализация всего, чего угодно (главный метод)
                public void initMain()
                {
                    //Инициализация списка отображения параметров
                    public void initListView(String idMaterial, String idCAE)
                    {
                        //Метод перевода параметров на английский для САЕ-
                        среды Solidworks
                    }
                }
            }
        }
    }
}
```

```
public void translatePropertyForSolidworks()
```

Алгоритм модуля представлен на рисунке 3.5.

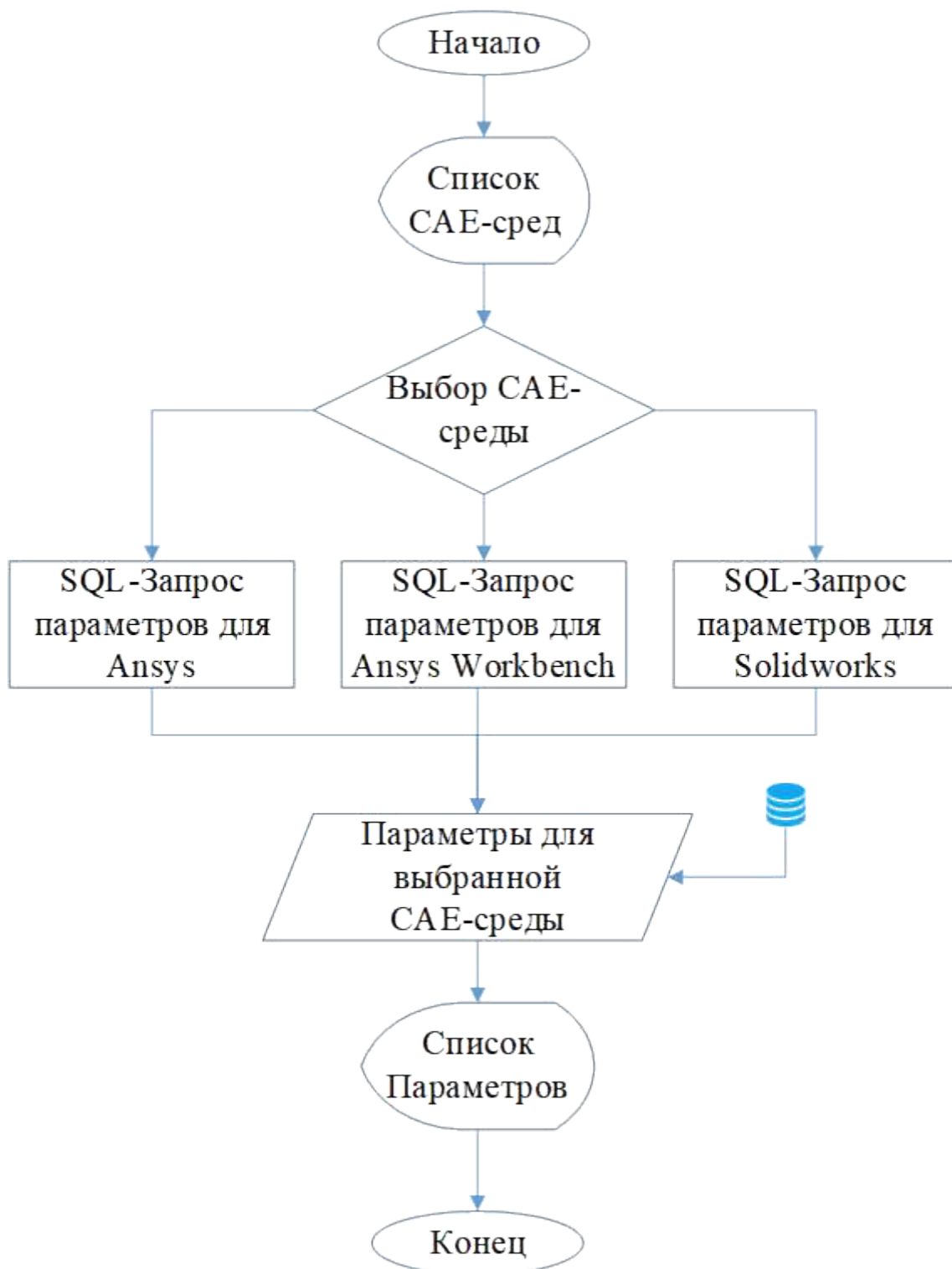


Рисунок 3.5 – Алгоритм модуля вывода параметров

Входные данные:

- id-значение материала
- id-значение САЕ-среды

Выходные данные:

- Названия для основных параметров
- Текстовые пояснения

- Единицы измерения для каждого параметра
- Значение для каждого параметра

Пример SQL-запроса в формате OC Android:

```
//Имена таблиц, к которым передается запрос
String table = "((PROPERTY_TABLE inner join UNITS_TABLE
on PROPERTY_TABLE.units_id = UNITS_TABLE.id_units)" +
"PROPERTY_TABLE inner join MECHANICAL_TABLE on
PROPERTY_TABLE.id_property =
MECHANICAL_TABLE.property_id)" + "PROPERTY_TABLE inner
join ANALYSIS_PROPERTY_TABLE on
PROPERTY_TABLE.id_property=
ANALYSIS_PROPERTY_TABLE.property_id";

//Список полей, которые мы хотим получить
String columns[] = {"id_property as _id",
"PROPERTY_TABLE.material_id as ID_PROPERTY_TABLE",
"UNITS_TABLE.material_id as ID_UNITS_TABLE",
"MECHANICAL_TABLE.material_id as ID_MECHANICAL_TABLE",
"name_property",
"name_units",
"value",
"caption_property"
};

//Параметр, формирующий выражение WHERE
String selection = "ID_MECHANICAL_TABLE = ? and cae_id
= ? or cae_id = ?";

//Массив аргументов для selection
String[] selectionArgs = {idMaterial, idCAE, idMaterial};

//Группировка
String groupBy = null;
```

```

        //Фильтр для группировки
String having = null;
        //Сортировка
String orderBy = "weight";
        //Запрос через объект класса Cursor
Cursor cursor = database.query(table, columns, selection,
selectionArgs, groupBy, having, orderBy);

```

3.3 Модуль создания нового материала

Данный модуль отвечает за реализацию такой функции, как создание нового материала в приложении. Минимальные входные параметры — это название создаваемого материала и значения для пяти основных параметров. На выходе же получаем готовый материал с пятью параметрами, единицами измерения для каждого параметра и три вариации представления параметров, по одной на имеющуюся CAE-среду в приложении. Помимо создания материала с нуля, модуль дает возможность копировать любой доступный в базе данных материал и расширить его нужными параметрами.

Данный модуль реализован в виде Java-класса и содержит следующие методы:

```

        //Создание экрана из xml-разметки
protected void onCreate(Bundle savedInstanceState)
        //Инициализация toolbar (панель инструментов)
public void initToolbar()
        //Инициализация всего, чего угодно (главный метод)
public void initMain()
        //Метод копирования параметров открытого материала
public void copyMaterial()
        //Метод сохранения материала
public void fabCreateComplete(View view)
        //Метод добавления нового параметра в материал

```



```
public void fabAddNewProperty(View view)
```

Алгоритм модуля изображен на рисунке 3.6.

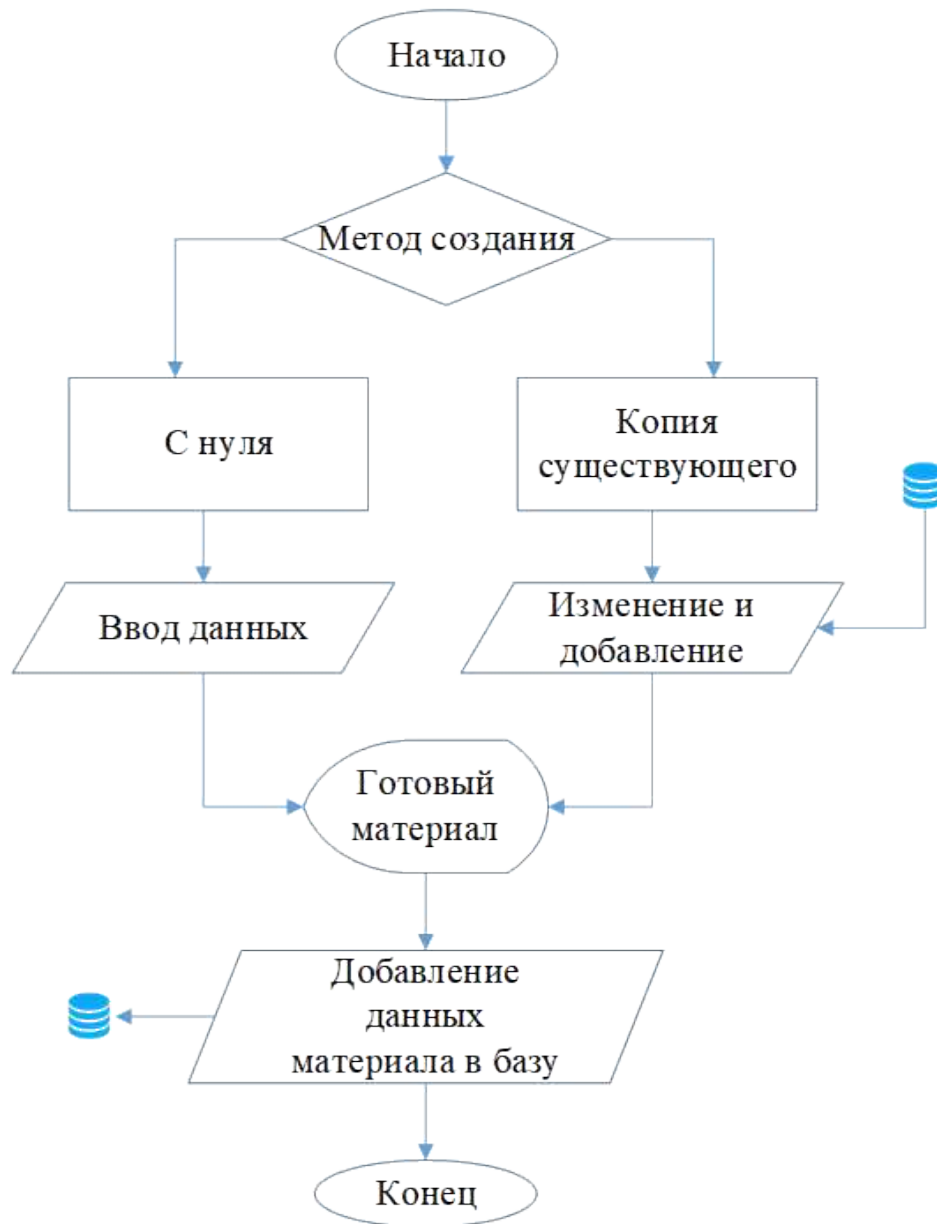


Рисунок 3.6 – Алгоритм модуля создания материала

Входные данные:

- Название материала
- Значения пяти основных параметров

Выходные данные:

- Готовый материал
- Единицы измерения для каждого параметра
- Три представления параметров, по одному для каждой САЕ-среды, имеющейся в приложении

Примеры SQL-запросов в формате OC Android:

```
//Добавление названия в MATERIAL_TABLE
```

```

ContentValues cvMATERIAL_TABLE = new ContentValues();
cvMATERIAL_TABLE.put("name_material",
edNameNewMaterial.getText().toString());
cvMATERIAL_TABLE.put("description_material", "users");
long rowID = db.insert("MATERIAL_TABLE", null, cvMATERIAL_TABLE);

    //Добавление значения Модуля Юнга в MECHANICAL_TABLE
ContentValues cvMECHANICAL_TABLE = new ContentValues();
cvMECHANICAL_TABLE.put("material_id", rowID);
cvMECHANICAL_TABLE.put("property_id", 1);
cvMECHANICAL_TABLE.put("value", Double.parseDouble(value_prop_1));
db.insert("MECHANICAL_TABLE", null, cvMECHANICAL_TABLE);

    //Добавление значения коэф. Пуассона в MECHANICAL_TABLE
cvMECHANICAL_TABLE.put("material_id", rowID);
cvMECHANICAL_TABLE.put("property_id", 2);
cvMECHANICAL_TABLE.put("value", Double.parseDouble(value_prop_2));
db.insert("MECHANICAL_TABLE", null, cvMECHANICAL_TABLE);

    //Добавление значения плотности в MECHANICAL_TABLE
cvMECHANICAL_TABLE.put("material_id", rowID);
cvMECHANICAL_TABLE.put("property_id", 3);
cvMECHANICAL_TABLE.put("value", Double.parseDouble(value_prop_3));
db.insert("MECHANICAL_TABLE", null, cvMECHANICAL_TABLE);

    //Добавление значения предела прочности в
MECHANICAL_TABLE
cvMECHANICAL_TABLE.put("material_id", rowID);
cvMECHANICAL_TABLE.put("property_id", 4);
cvMECHANICAL_TABLE.put("value", Double.parseDouble(value_prop_4));
db.insert("MECHANICAL_TABLE", null, cvMECHANICAL_TABLE);

    //Добавление значения предела текучести в
MECHANICAL_TABLE
cvMECHANICAL_TABLE.put("material_id", rowID);
cvMECHANICAL_TABLE.put("property_id", 5);
cvMECHANICAL_TABLE.put("value", Double.parseDouble(value_prop_5));
db.insert("MECHANICAL_TABLE", null, cvMECHANICAL_TABLE);

```

3.4 Модуль работы с созданными материалами

Этот модуль отвечает за отображение и работу только с созданными пользователем материалами. Некое хранилище, где можно переименовать или удалить созданные материалы. Для отображения пользовательских материалов модуль принимает значение поля `description_material` из таблицы `MATERIAL_TABLE`, в том случае, если оно равно значению «users», то sql-запрос берет данный материал и отображает его в хранилище «Мои материалы».

Данный модуль реализован в виде Java-класса и содержит следующие методы:

```
//Создание экрана из xml-разметки
protected void onCreate(Bundle savedInstanceState)
    //Инициализация toolbar (панель инструментов)
public void initToolbar()
    //Внутренний класс-адаптер, организующий
представление и вывод созданных пользователем материалов
public class MySimpleCursorAdapter extends
SimpleCursorAdapter
    //Инициализация списка отображения параметров
public void initListView()
    //Метод появления всплывающего окна для выбора CAE-
среды
public void showAlertCAE(Context context, final String
itemName, final String itemNameID)
    //Метод, который открывает окно для отображения
параметров материала
public void openPropertyActivity(String nameMaterial,
String nameCAE, String idMaterial, String idCAE)
    //Метод переименования материал
```

```

public void renameItem(String itemName, final String
itemID)

    //Метод удаления материала
public void deleteItem(String itemName, final String
itemID)

```

Алгоритм модуля представлен на рисунке 3.7.

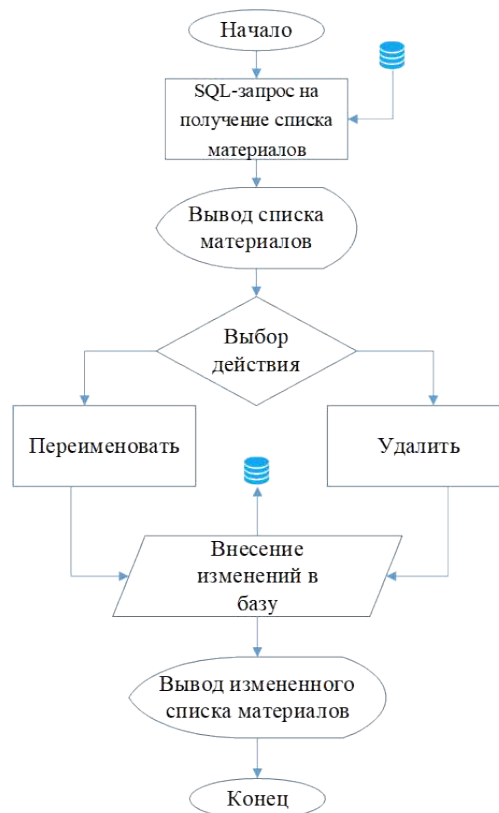


Рисунок 3.7 – Алгоритм модуля работы с созданными материалами

Входные данные:

- Значение поля description в таблице MATERIAL_TABLE

Выходные данные:

- Материалы

Примеры SQL-запросов в формате OC Android:

```

"SELECT id_material as _id, name_material FROM
MATERIAL_TABLE WHERE description_material = 'users'"

```

3.5 Модуль добавления параметров

Этот модуль отвечает за возможность расширять создаваемый материал параметрами помимо пяти основных. Модуль принимает на вход три

параметра: название свойства, его единицы измерения и значение. На выходе модуль создает готовое свойство, которое становится частью материала.

Данный модуль реализован в виде Java-класса и содержит следующие методы:

```
//Метод появления окна для заполнения значений  
создаваемого параметра  
public void fabAddNewProperty(View view)
```

Алгоритм модуля добавления параметра изображен на рисунке 3.8.

Входные данные:

- Название параметра
- Единицы измерения
- Значение параметра

Выходные данные:

- Готовое свойство с тремя единицами данных

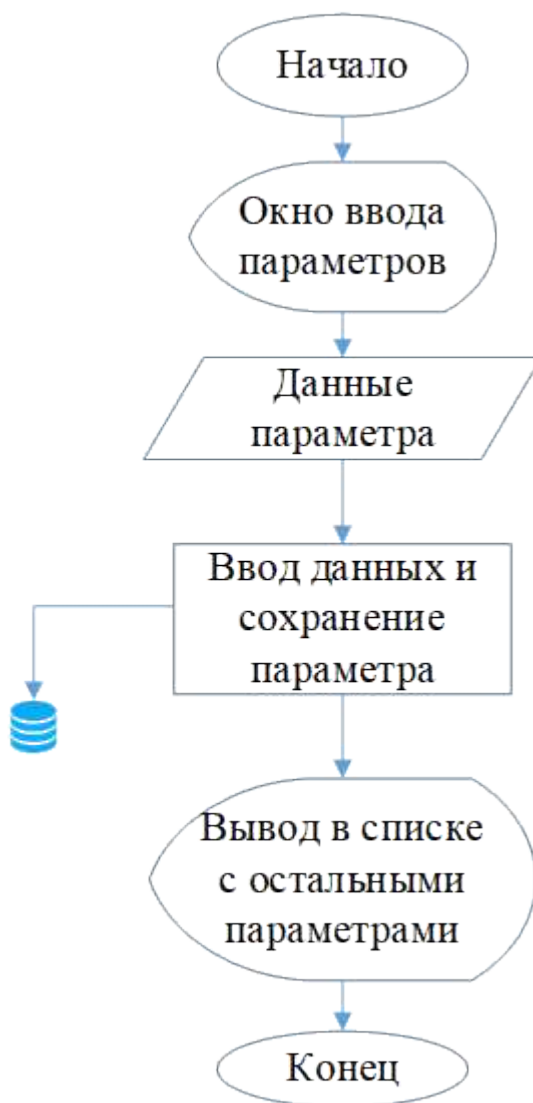


Рисунок 3.8 – Алгоритм модуля добавления параметра

Примеры SQL-запросов в формате OC Android:

```

//Внес единицы измерения в таблицу UNITS_TABLE
ContentValues cvUNITS_TABLE = new ContentValues();
cvUNITS_TABLE.put("material_id", rowID);
cvUNITS_TABLE.put("name_units", ((TextView)
arrList_tvUnits.get(i)).getText().toString());
long idRowUNITS_TABLE = db.insert("UNITS_TABLE", null,
cvUNITS_TABLE);

```

```

//Внес название параметра в таблицу PROPERTY_TABLE
ContentValues cvPROPERTY_TABLE = new ContentValues();
cvPROPERTY_TABLE.put("material_id", rowID);

```

```

cvPROPERTY_TABLE.put("name_property", ((TextView)
arrList_tvCaption.get(i)).getText().toString());
cvPROPERTY_TABLE.put("units_id", idRowUNITS_TABLE);
long idRowPROPERTY_TABLE = db.insert("PROPERTY_TABLE", null,
cvPROPERTY_TABLE);

    //Внес id параметра и значение параметра в таблицу
MECHANICAL_TABLE
ContentValues cvMECHANICAL_TABLE_users = new ContentValues();
cvMECHANICAL_TABLE_users.put("material_id", rowID);
cvMECHANICAL_TABLE_users.put("property_id", idRowPROPERTY_TABLE);
cvMECHANICAL_TABLE_users.put("value", ((EditText)
arrList_editValue.get(i)).getText().toString());
db.insert("MECHANICAL_TABLE", null, cvMECHANICAL_TABLE_users);

    //Внес id-параметра и описание свойства в таблицу
ANALYSIS_PROPERTY_TABLE
ContentValues cvANALYSIS_PROPERTY_TABLE = new ContentValues();
cvANALYSIS_PROPERTY_TABLE.put("property_id", idRowPROPERTY_TABLE);
cvANALYSIS_PROPERTY_TABLE.put("cae_id", rowID);
cvANALYSIS_PROPERTY_TABLE.put("caption_property", ((TextView)
arrList_tvCaption.get(i)).getText().toString());
cvANALYSIS_PROPERTY_TABLE.put("weight", 6 + i);
db.insert("ANALYSIS_PROPERTY_TABLE", null,
cvANALYSIS_PROPERTY_TABLE);

```

3.6 Тестирование приложения

Данный раздел освещает тестирование приложения на основе создания тестовых кейсов. На основе которых были разработаны проверки и методы защиты приложения от критических ошибок и некорректного выполнения работы. Некоторые примеры подобных кейсов приведены ниже.

Пример 1.

При создании материала нам необходимо ввести его название. Это одно из основных свойств материала, без которого приложение не сможет с ним

взаимодействовать, например, добавить в базу данных или вывести в поиске. До проведения тестирования, это никак не регулировалось, то есть пользователь мог попросту забыть ввести название, сохранить материал, а дальше все попытки найти его среди прочих были бы безуспешны, так как нет названия, по которому приложение смогло бы вывести его при поиске.

Решением этой проблемы было прописать проверку на пустоту EditText-у в которое прописывается имя материала. То есть, пока это поле пустое – приложение не позволит вам сохранить созданный материал.

Реализация:

```
//Проверка edNameNewMaterial на пустоту
if(!edNameNewMaterial.getText().toString().equals("")){
    //Код метода
}
```

Пример 2.

По тому же принципу была предотвращена возможность ошибки при переименовании созданных пользователем материалов. Проблема полностью аналогично, поэтому подробно описывать нет необходимости.

Реализация:

```
//Проверка alertEdit_myMaterial на пустоту
if(!alertEdit_myMaterial ().toString().equals("")) {
    //Код метода
}
```

Пример 3.

Еще одна из возможных пользовательских ошибок, это случайно или как-то иначе, при создании материала, в поле, предназначенное для числового значения, ввести текст. Данная ошибка приведет к критической ошибке приложения и закроет его, так данные из этого поля заносятся в столбец в базе данных у которого тип данных INTEGER. При попытке sql-запроса вставить

текстовое значение, SQLite определил бы несоответствие требуемых и предоставленных данных и завершил приложение в ошибке.

Чтоб исключить возможность таких случайностей, было назначено для поля, куда требуется вводить числовое значение, доступ только к числовой клавиатуре. То есть при установке курсора в данный EditText клавиатура либо сразу появляется только в числовом формате, либо меняется на таковой, если до этого открыта в текстовом формате.

Реализация:

```
<EditText
    android:id="@+id/edValueNewProperty"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:ems="10"
    android:gravity="center_horizontal"
    android:hint="- , --"
    android:inputType="numberDecimal"
    android:textColor="#99ffffff" />
```

ВЫВОДЫ

Подводя итоги проделанной работы можно сказать, что все поставленные задачи, среди которых:

- Создание и нормализация базы данных.
- Добавление не менее 200 единиц статей в базу.
- Реализация модуля создания нового материала.
- Реализация модуля создания параметров отсутствующих в базе.
- Реализация модуля вывода параметров для разных САЕ-сред.
- Реализация модуля копирования материала из базы и его настройка/редактирование.
- Разработка дизайна приложения.
- Представление готового и полностью рабочего приложения для операционной системы Android.
- Тестирование готового приложения.

были выполнены в полном объеме.

Таблица 3.1 – Спецификация приложения «Справочник материалов»

Параметр приложения	Пояснение
Операционная система	Android
Требуемая версия API/Android	от версии 16 / 4.1-Jelly Bean и выше
Локализация	Полностью русское
Доступ	Свободный доступ Полностью бесплатное
Место, занимаемое на устройстве	4 Мб
Распространение	Google Play Маркет

Спецификация разработанного приложения представлена в таблице 3.1

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Описание термина «CAE-системы» [Электронный ресурс]: Systems and software engineering . Режим доступа: <http://sewiki.ru/CAE-%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0>
2. Google [Электронный ресурс]. Режим доступа: <https://ru.wikipedia.org/wiki/Google>
3. Описание операционной системы Android [Электронный ресурс]: Википедия – свободная энциклопедия, которую можно редактировать. Режим доступа: <https://ru.wikipedia.org/wiki/Android>
4. API операционной системы Android [Электронный ресурс]: Хабрахабр – многофункциональный сайт, представляющий собой смешение новостного сайта и коллективного блога. Режим доступа: <https://ru.wikipedia.org/wiki/Android>
5. Процентная статистика использования версий Android [Электронный ресурс]: Android Developers. Режим доступа: <https://developer.android.com/about/dashboards/index.html>
6. Реляционные базы данных [Электронный ресурс]: Amazon relational database. Режим доступа: <https://aws.amazon.com/ru/relational-database/>
7. Нормализация базы данных [Электронный ресурс]: Microsoft - кампания по разработке программного обеспечения. Режим доступа: <https://support.microsoft.com/ru-ru/help/283878/description-of-the-database-normalization-basics>
8. SQLite базы данных [Электронный ресурс]: «SoftTime» - IT-студия. Режим доступа: http://www.softtime.ru/info/article.php?id_article=70
9. Среда разработки «Android Studio» [Электронный ресурс]: «WinFox» - разработчики мобильных приложений. Режим доступа: <http://wnfx.ru/android-studio-ide-ot-google/>
10. Приложение-аналог №1 «Сплавы металлов. Марочник» [Электронный ресурс]: «Google Play» - магазин приложений, игр, книг, музыки и фильмов компании Google. Режим доступа: <https://play.google.com/store/apps/details?id=ru.channelplus.metals>
11. Приложение-аналог №2 «Сплавы и Металлы» [Электронный ресурс]: «Google Play» - магазин приложений, игр, книг, музыки и фильмов компании Google. Режим доступа: <https://play.google.com/store/apps/details?id=ua.at.alloysandmetals>
12. Приложение-аналог №3 «Справочник инженера сварщика» [Электронный ресурс]: «Google Play» - магазин приложений, игр, книг, музыки и фильмов компании Google. Режим доступа: <https://play.google.com/store/apps/details?id=ru.weldword.android.eshpweldhandbook&hl=ru>

13. Жизненный цикл Activity в Android [Электронный ресурс]: «Developer Android» - разработчики мобильных приложений. Режим доступа:
<https://developer.android.com/guide/components/activities.html#Lifecycle>
14. Модульное тестирование [Электронный ресурс]: «Software-testing». Режим доступа: <http://software-testing.ru/library/testing/functional-testing/77-unit-testing>

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Политехнический институт
институт
Конструкторско-технологическое обеспечение машиностроительных
производств
кафедра

УТВЕРЖДАЮ

Заведующий кафедрой

М.П. Головин

инициалы, фамилия

подпись

«12» 06 2018 г.

БАКАЛАВРСКАЯ РАБОТА

09.03.01 Информатика и вычислительная техника
код – наименование направления

«Мобильное приложение для управления базой данных материалов»
тема

Руководитель

02.06.18
подпись, дата

ст. преподаватель КТОМП
должность, ученая степень

Д. И. Морозов
инициалы, фамилия

Выпускник

02.06.18
подпись, дата

Е. Е. Куксов
инициалы, фамилия

Красноярск 2018